

文章编号:1001-9081(2008)06-1454-03

一种高效的图像数据库检索方法

唐 敏^{1,2}, 阳爱民^{1,2}

(1. 湖南工业大学 计算机与通信学院,湖南 株洲 412008; 2. 国防科学技术大学 计算机学院,长沙 410073)
(monicabey@126.com)

摘要:对于大型图像库,如何高效地检索出相似图像是图像检索系统的一大挑战。提出了一种改进的 K-均值聚类算法建立分层结构的索引,再利用 A*树算法和三角不等式原则及 N 近邻方法对索引库快速高效地搜索,达到对图像库快速高效检索相似图像的目的。实验在 Corel 图像库上进行,实验结果表明该方法以对数时间复杂度实现基于内容的高效检索。

关键词:K-均值聚类算法;A*树算法;三角不等式原则;N近邻方法

中图分类号: TP391.3 文献标志码:A

Efficient CBIR retrieval method for image database

TANG Min^{1,2}, YANG Ai-min^{1,2}

(1. College of Computer and Communication, Hunan University of Technology, Zhuzhou Hunan 412008, China;
2. College of Computer, National University of Defense Technology, Changsha Hunan 410073, China)

Abstract: How to retrieve similar images from large images database efficiently is a great challenge for content-based image retrieval system. A modified K-means algorithm was proposed to form the hierarchy of the indexing structure. A * search algorithm, triangle inequality principle and N-near neighbours were applied to achieve an optimal search in order to retrieve efficiently for the large image database. Experiments on Corel database show that the proposed algorithm achieves efficient logarithm retrieval.

Key words: K-means cluster algorithm; A* tree algorithm; triangle inequality principal; N-near neighbours method

0 引言

随着图像库容量的迅速增大,图像检索的实时性要求日益突出。检索系统对查询输入的响应时间成为图像检索系统能否实用化的关键。在应用中,图像数据库耗时较大的操作有两方面:特征提取和相似度计算。特征提取只在建立图像特征库时才计算,相似度量则是每一次 QBE (Query-By-Example) 查询都需要计算。在大型图像库中进行相似查询是十分艰巨的任务,图像内容特征通常都是高维的矢量,在多特征联合检索中,特征矢量甚至达到一百维以上。传统的图像查询采用顺序扫描方法 (Sequential Scan Algorithm, SSA),该方法距离计算的花费很大,检索效率低,因而建立合适的索引机制来提高检索效率是十分必要的。比较流行的高维索引方法有 R 树、Quad 树以及 Grid 树。但是当向量维数超过 20 时,以上方法的效率均迅速降低,几乎等价于顺序查找。针对以上索引方法的局限性,有人提出了将聚类用于索引的思想。这种索引方法通过无监督的学习将特征空间分类,相似的类别拓扑相近,适合最近邻搜索,大大减少了搜索的次数。

本文应用已有的研究基础完成特征提取,建立图像特征库。然后提出一种改进的 K-均值聚类算法对图像特征库建立分层结构的索引,再提出一种新的搜索机制,实现对大型图像库的高效检索。

1 特征提取和相似性计算

本文是在研究图像特征提取和相似性计算方法^[1-2]的基

础上进行的继续研究。由于特征提取和相似性计算方法的确定将直接影响检索的性能,所以下面将简要介绍文献[1-2]中的特征提取过程和相似性计算方法。

1.1 特征提取

在文献[1-2]中,特征提取是先将图像分割成区域,流程如图 1 所示。然后对区域特征进行模糊化,最后得到区域模糊特征。由于模糊集合理论脱离了精确计算,符合人类检索图像的需要,所以引入了模糊集理论。应用计算简单的 Cauthy 函数作为隶属度函数 μ ,作用于各区域,得到区域的模糊特征,模糊颜色直方图、模糊纹理特征和模糊形状特征,分别按式(1)~(3)计算:

$$h_{n1} = \sum_{c' \in U^c} h_{n1}(c')\mu_c(c') \quad (1)$$

$$\vec{f}_n^T = \sum_{f_j \in U^S} f_j^T \mu_n(f_j) \quad (2)$$

$$\vec{f}_n^S = \sum_{f_j \in U^S} f_j^S \mu_n(f_j) \quad (3)$$

对大型图像库中的每幅图像实现特征提取,建立图像特征库。

1.2 相似度计算

最后对一幅图像的各个区域得到的特征有:权重(区域所占该图像面积的比例)、模糊颜色直方图、模糊纹理特征、模糊形状特征。由于一幅图像中的一个区域可能对应另一幅图像中的多个区域,所以采用区域集成匹配的相似性算法^[1-2]步骤如下(假设 Image1 有 M 个区域,Image2 有 N 个区域):

收稿日期:2007-12-22;修回日期:2008-02-04。

基金项目:中国博士后科学基金资助项目(20070410299);湖南省教育厅科研项目(07B014)。

作者简介:唐敏(1983-),女,湖南株洲人,硕士研究生,主要研究方向:智能信息处理、图像检索; 阳爱民(1970-),男,湖南永州人,教授,研究员,博士,主要研究方向:智能信息处理、图像检索、网络流量分类、文本分类。

1) 计算 Image1 中的一个区域和 Image2 的所有区域的距离, 取最小值, 以 i 为例有式(4):

$$R_{i, \text{Image}2} = \min_{1 \leq j \leq N} DIST(i, j) \quad (4)$$

2) 计算 Image2 中的一个区域和 Image1 的所有区域的距离, 取最小值, 以 j 为例有式(5):

$$R_{j, \text{Image}1} = \min_{1 \leq i \leq M} DIST(j, i) \quad (5)$$

3) 在获取 $M + N$ 个距离之后, 用式(6)计算 Image1 和 Image2 之间的距离:

$$Dist(1, 2) = \left(\sum_{i=1}^M \omega_{1i} R_{i, \text{Image}2} + \sum_{j=1}^N \omega_{2j} R_{j, \text{Image}1} \right) / 2 \quad (6)$$

其中 ω_{1i}, ω_{2j} 分别为 Image1 和 Image2 的区域 i 和区域 j 的权重。

4) 通过式(7)将距离转为相似度:

$$\text{sim}(1, 2) = \beta / (Dist(1, 2) + \beta) \quad (7)$$

其中 β 是一个经验参数, 是经过反复实验得出的一个经验值。

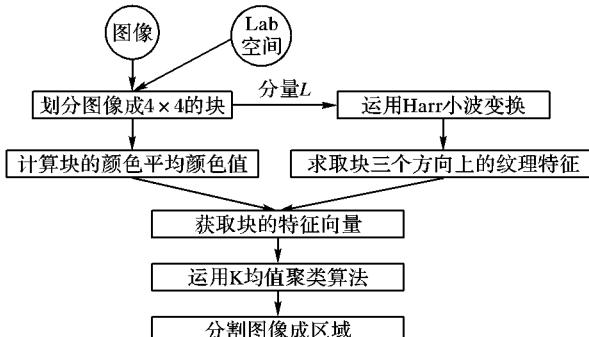


图1 图像分割流程

2 分层结构的索引

第1章对大型图像库建立的特征库将十分庞大, 如果用顺序方法检索, 效率将十分低下。本文针对庞大的特征库, 提出了一种改进的聚类算法建立索引, 实现高效检索。该算法的主体仍然采用 K-均值, 这是因为 K-均值聚类算法在处理大数据集方面具有其他聚类算法所无法比拟的优势。同时针对 K-均值聚类算法的缺陷, 诸如聚类个数以及模板初值的选定等问题, 进行了相应的改进, 使得这些经验值可以通过初始聚类动态地生成, 无需用户事先指定。

整个算法大致可以分为两个主要步骤, 即初始聚类与 K-均值迭代。

2.1 初始聚类

这个阶段的主要工作在于给出图像特征库一个大体的分组, 从而确定出较为合理的聚类个数以及模板初值, 为接下来的 K-均值迭代提供必要的经验值。该过程的实质在于寻找初始聚类中心。具体方法如下: 首先计算图像特征库中所有特征向量之间的距离。由于噪声干扰, 过滤掉总距离数的 $1/10$ 的最大距离。对剩下的距离求平均, 并赋予 rad_1 ; 同时, 取 $rad_2 = rad_1/\sigma (\sigma \geq 1)$ 。然后以每个区域为中心, 以 rad_1 为特征分组的临界距离, 计算落在该组内的特征数目, 将按组内特征数目从大到小排序, 取最大者为第一个初始聚类中心 C_1 ; 若次大者与第一个中心之间距离大于 rad_2 , 则将之作为第二个中心 C_2 , 否则继续判定下一最大者; 若下一最大者与前面若干个中心之间距离均大于 rad_2 , 则将之作为又一新中心。以此类推, 最终可以得到一组初始聚类中心 $\{C_1, C_2, \dots, C_k\}$ 。

2.2 K-均值迭代

经过初始聚类, 得到聚类数目 K 以及模板初值 $C = \{C_1, C_2, \dots, C_k\}$, 接下来将进行 K-均值迭代。其基本原理是根据库中所有区域特征与聚类中心距离的远近程度, 形成 K 个互

不相交的聚类, 较为相似的都聚在同一类中。对特征库中所有特征向量, 运用相同的改进 K-均值聚类算法, 递归地形成分层结构的索引。索引的深度基于图像特征库大小自适应确定。

3 搜索算法及相似排序

检索过程实际包括: 1) 对已有索引的搜索过程; 2) 对搜索到的相似图像进行相似度计算, 并按相似度排序。

3.1 搜索算法

一个好的搜索算法能够快速地找到相似图像而不漏掉相似图像。为达到最优搜索, 我们运用 A* 搜索算法^[3], 其通过保存已经被搜索过的所有节点的路径, 选择对查询区域有最小距离的节点。

为防止漏检图像, 本文在运用 A* 算法选择路径分支时, 采用 N 近邻方法^[4]选择前 N 个最小分支。

在图像库中, 一般与叶子节点相关的图像集合远小于原始图像集合。减少的图像集合可以通过三角不等式原则过滤掉。如果两个区域 p 和 r 分别对应于索引中的一个叶子节点和查询图像的一个区域, 则判断该叶子节点对应的图像是否与查询图像相似, 由式(8)决定。如果成立, 则表示两幅图像相似, 保留该节点以便寻找到对应的相似图像; 如果不成立, 则过滤掉该节点。

$$DIST(p, r) \geq |DIST(p, z) - DIST(r, z)| \quad (8)$$

其中 $DIST(p, r)$ 是区域 p 和 r 的距离, z 表示对应叶子节点(聚类)的中心。

因此, 我们的检索方法是先用改进的 K-均值聚类算法建立分层索引的, 然后用三角不等式过滤并采用 N 近邻的 A* 树的方法, 简称为 KTNM, 并有对数复杂度检索的结论。

在 KTNM 算法(如算法 1 所示)中使用的标记有: m 是查询图像的区域数目, s^* 是聚类, 其中心到查询区域 r 有最小距离, Ω 是已经搜索到的聚类集合, $|c(s^*)|$ 是 s^* 的孩子子集的大小, z_s 是聚类中心, y 是查询区域与聚类中心距离的排序, N 表示 N 近邻, 结果 ψ 是最终图像集。

输入: 查询图像 Q , 预定义阈值 t ;

输出: 用查询图像 Q 检索出的图像集 ψ 。

begin

//查询图像 Q 的分割:

1) 块特征提取如第1章所述, 得到块特征向量 $[b_i^{C1}, b_i^{C2}, b_i^{C3}, b_i^{T1}, b_i^{T2}, b_i^{T3}]^T$;

2) K-均值聚类, 分割图像得 m 个区域;

3) 区域形状特征提取, 运用 1~3 阶的归一化惯量, 得到区域的特征向量 $[R_r^{C1}, R_r^{C2}, R_r^{C3}, R_r^{T1}, R_r^{T2}, R_r^{T3}, R_r^{S1}, R_r^{S2}, R_r^{S3}]^T$;

4) 区域特征模糊化, 利用模糊集合理论, 得到区域模糊特征向量;

//到索引库中找到相关图像的搜索过程:

for 对查询图像 Q 的每个区域 $r_i (1 \leq i \leq m)$

$s^* = \text{root};$

$\Omega = \{s^*\}$;

While s^* 不是叶子节点的上层节点

$\Omega \leftarrow (\Omega - \{s^*\}) \cup c(s^*)$;

$y \leftarrow \text{sort}(DIST(r, z_s))$;

for 对 N 近邻的 $k (1 \leq k \leq N)$

$s^*(k) = \Omega(y(k))$;

end

end

$\Phi = \{\}$;

for 在节点 s^* 的每个区域 p

$if |Dist(p, z_s) - DIST(r, z_s)| \leq t$

$\Phi \leftarrow \Phi \cup \{p\}$

```

end
 $\psi_r = \{\text{包含 } \Phi \text{ 中区域的图像集合}\};$ 
end
end
 $\psi = \bigcup_{r=1}^m \psi_r$ 
end

```

结论 在平均情况下, KTNM 算法达到对数的检索效率。

证明 假设 f 是索引的平均分支因子, n 为数据库的图像数目, g 是一幅图像区域的平均数目, d 是索引树的高度, 则 ng 是区域总数。在平均情况下, 当 $d \geq \log_f(ng) - \log_f(\log_f(ng))$ 时, 一个叶子节点的中心对应的区域数为 $\omega \leq \log_f(ng)$ 。在已选择的叶子节点中心 s^* 中, 运用三角不等式原则。在没有损失的情况下, 假设 s^* 对应的区域集合中的一个区域与 s^* 的比为 λ , λ 是 s^* 对应区域集合的半径, 并运用三角不等式原则保留那些半径为 λ/τ 的区域, 其中 $1/\tau$ 是阈值。结果, 用来与查询区域比较的区域的平均数目 $q = \omega/\tau^2 \leq \log_f(ng)/\tau^2$, 称这些区域为候选区域。因为一个候选区域至多对应数据库中的一幅图像, 所以图像库中的总图像至多是 $qg = g \cdot \log_f(ng)/\tau^2 = g/\tau^2 \cdot (\log_f n + \log_f g)$ 。因为 g 是一常量, 由图像分割时决定, 且 $g \ll n$, 所以有 $qg = O(\log_f n)$ 。这表示对 n 幅图像的数据库, 搜索算法 KTNM 的时间复杂性为 $O(\log_f n)$ 。

3.2 相似排序

上节中得到的相似图像集合 ψ 的图像数目要比图像库中的图像数 n 小得多, 因此检索效率得到提高。接下来要计算查询图像与 ψ 中的每幅图像的相似度, 并按照相似度从大到小进行排序, 然后依次返回相似度大于阈值的图像给用户。相似度计算采用 1.2 节中提到的集成区域匹配方法。而顺序查找是要对 n 幅图像计算相似度并排序。本文将查找时间定义为查找到相似图像所需的时间与对这些查找到的图像排序所需的时间之和。SSA 和 KTNM 的区别就在于进行相似度计算图像的数目。由此可以得出前面证明的查找时间为对数复杂度是正确的。

4 实验分析

4.1 实验结果

为了验证算法对大型图像库进行基于内容的检索是否有效, 我们建立了测试模型。图像数据库是含有 10 000 幅图像的 Corel, 包含风景、建筑、机动车辆等不同的类别, 且每个类别的图像数 25~200 不等。本系统从图像库中随机选取 50 幅作为例子图像集合。系统在 P4 Celeron(r) 1.70 GHz, 512 MB 内存、Windows XP 环境下, 用 Matlab7.0 实现的。比较顺序查找、单纯的 K-均值索引算法和本系统检索方法 KTNM 的平均返回结果的时间和返回图像的准确性。

1) 以下是在 243 幅图像库上实验的部分数据, 包括查到的图像数、查到的相关图像数、查准率、查全率及查找时间。查准率与查全率的定义如下:

查准率 = 查到的相关图像数 / 查到的图像数

查全率 = 查到的相关图像数 / 库中实际相关的图像数

(1) 阈值: 相似度 ≥ 0.5 , 测试结果如表 1 所示。

表 1 在 243 幅图像库上相似度为 0.5 的测试结果

算法	查到的图像数	查到的相关图像数	查准率/%	查全率/%	查找时间/s
顺序查找	90	50	55.56	90.91	10.78
K-均值	72	47	65.28	85.45	7.11
KTNM	55	44	80.00	80.00	3.32

(2) 阈值: 相似度 ≥ 0.3 , 测试结果如表 2 所示。

表 2 在 243 幅图像库上相似度为 0.3 的测试结果

算法	查到的图像数	查到的相关图像数	查准率/%	查全率/%	查找时间/s
顺序查找	137	53	38.69	96.36	19.17
K-均值	87	49	56.32	89.09	14.28
KTNM	63	46	73.02	83.64	8.16

测试结果表明: 采用 KTNM 方法, 图像查准率相比于顺序查找有明显提高, 查找时间较顺序查找的短。同时, 在查准率和查找时间上, 该算法均优于 K-均值算法。由此可见, 采用 KTNM 方法进行检索, 可以提高检索效率, 而且相似度越低, 索引效果越明显。

2) 以下是在 10 000 幅的图像库上实验的部分数据。阈值: 相似度 ≥ 0.5 , 测试结果如表 3 所示。

对于大型图像库而言, 很难找出与例子图像相关的所有图像数目。因为“相关”的概念是很模糊的, 人们从肉眼上直观感觉到的相关却很难让机器去识别。定义查到率 = 查到的图像数 / 顺序查到的图像数, 并认为顺序查找的查到率为 100%。表 3 的测试结果表明: 采用 KTNM, 查找例子图像的时间缩短为顺序查找的近 1/5, 这个结果基本上可以满足用户大容量检索的需求。

表 3 在 10 000 幅图像库上相似度为 0.5 的测试结果

算法	查到的图像数	查到率/%	查找时间/s
顺序查找	429	100.00	889.58
K-均值	314	73.19	626.50
KTNM	181	42.19	178.32

4.2 实验分析

用索引之后, 图像的平均查找次数为: $t1 = clr + n/clr$ 。其中 clr 为索引结构中聚类中心的个数, n 为图像库中图像数目。而顺序查找的平均查找次数为 $t2 = n$ 。对于大型图像库而言, $clr \ll n$, 因此会有 $t1 \ll t2$, 即采用索引之后图像的平均查找次数会大大降低, 从而使得查找到相似图像的时间大大减少。

通过在大型图像库上对算法进行反复测试, 不论是采用顺序查找, 还是采用 KTNM 方法进行查找, 其查找时间都随着相似度的升高而降低。这是因为随着用户要求的相似度的升高, 系统能够返回的图像数目必然也会减少, 图像按相似度大小排序的时间随之减少, 从而使得查找时间降低。随着相似度的降低, KTNM 检索效率越为明显, 即相似度越低时, KTNM 相对顺序查找和 K-均值查找在高相似度情况下更有优势, 然而与此同时图像的查准率也随之降低。可见, 图像的查准率与查找时间是互为矛盾的, 要想减少查找时间, 需要以牺牲查准率作为代价。

5 结语

本文介绍了图像特征提取方法和相似度计算方法, 提出了一种新的改进的 K-均值聚类算法建立分层结构的索引和一种新的搜索机制。实验结果表明, 结合分层结构的索引和搜索机制的检索方法能提高检索效率, 实现快速高效的基于内容的图像检索。

但是算法中也存在一些问题有待改进: 实验发现, 图像的查准率与查找时间互为矛盾, 应设法寻找二者的性能最优点, 使用用户的需求尽可能地得到满足。还需要一个更快的排序算

(下转第 1459 页)

```

 $\Theta = \Theta^-(a);$ 
更新  $\Theta$  中各属性的频度信息;
if  $CORE(\Theta) \neq \emptyset$  then  $a = CORE(\Theta)$ 
else  $a = \Theta$  中频度最高的属性(若多个时任选一个);
}
return  $T$ ;
}

```

5) 约简树根节点到叶节点的每条路径所有属性对应一个约简,依次输出各条路径的对应属性集即得到信息系统 IS 的所有约简。

3 实例分析

本章用实例进一步说明基于差别矩阵的约简树构造方法。给定一决策信息系统 IS 如表 1,其中 $C = \{a, b, c, d, e, f\}$ 。

信息系统 IS 的正域等价类为 $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4, x_9\}, \{x_5, x_{10}\}$; 条件等价类为 $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4, x_9\}, \{x_5, x_{10}\}, \{x_6, x_7, x_8\}$ 。因此其差别矩阵的行简化为正域的条件等价类:

表 2 决策系统 IS 的改进差别矩阵

$[x_1]_{IND(C)}$	$[x_2]_{IND(C)}$	$[x_3]_{IND(C)}$	$[x_4]_{IND(C)}$	$[x_5]_{IND(C)}$	$[x_6]_{IND(C)}$
$[x_1]_{IND(C)}$					$\{f\}$
$[x_2]_{IND(C)}$	$\{a, b, d\}$				$\{a, b, d, f\}$
$[x_3]_{IND(C)}$	$\{a, d, e\}$	$\{b, e\}$			$\{a, d, e, f\}$
$[x_4]_{IND(C)}$	$\{e, f\}$	$\{a, b, d, e, f\}$	$\{a, d, f\}$		$\{e\}$
$[x_5]_{IND(C)}$	$\{a, c, e, f\}$	$\{b, c, d, e, f\}$	$\{c, d, f\}$	$\{a, c\}$	$\{a, c, e\}$

通过计算可得差别集 $\Theta = \{\{e\}, \{f\}, \{a, b, d\}, \{a, c\}\}$ 。以下详述利用函数 Tree_con_kernel 构造决策系统约简树的具体过程: 根节点 $root = CORE(\Theta) = \{\{e\}, \{f\}\}$; 更新差别集 $\Theta = \Theta^-(CORE(\Theta)) = \{\{a, b, d\}, \{a, c\}\}$; 计算属性的频度信息 $F(a, b, c, d) = (2, 1, 1, 1)$; 将属性频度最大的 a 作为根的第一棵子树; $\Theta' = \Theta - \{\Theta_i : \{a\} \cap \Theta_i \neq \emptyset\} = \emptyset$, 所以 a 即为叶节点; 构造根的第二棵子树时更新差别集 $\Theta = \Theta^-(a) = \{\{b, d\}, \{c\}\}$; 此时 $CORE(\Theta) = \{\{c\}\}$, 将其作为第二棵子树的根; $\Theta' = \Theta - \{\Theta_i : \{c\} \cap \Theta_i \neq \emptyset\} = \{\{b, d\}\}$; $\Theta = \Theta^-(c) = \{\{b, d\}, \{\emptyset\}\}$, 已包含元素 $\{\emptyset\}$, 第二层子树的根节点确定完成; 第二层第二棵子树的子树分别为 $\{b\}, \{d\}$ 。最后形成约简树见图 1, 从根节点到叶节点的所有路径 $\{\{efa\}, \{efcb\}, \{efcd\}\}$ 即对应决策系统的全部约简。

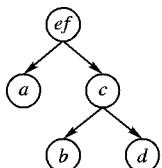


图 1 决策系统 IS 的约简树

(上接第 1456 页)

法以使查找到的大量图像按照相似度从大到小的顺序快速地返回给用户。算法在执行过程中需要较大的内存空间,应设法改进算法的执行效率,减少内存的使用。用户在进行检索时,有可能要求同时进行关键字检索和基于内容检索。由于传统数据库采用的是基于关键字的索引方法,而图像数据库采用的是高维特征的索引方法,两者之间存在很大差别,如何将两种索引结构结合起来,以适应各种用户不同的查询需求,将成为未来研究的一个方向。

参考文献:

$[x_1]_{IND(C)}, [x_2]_{IND(C)}, [x_3]_{IND(C)}, [x_4]_{IND(C)}, [x_5]_{IND(C)}$; 列简化为条件等价类: $[x_1]_{IND(C)}, [x_2]_{IND(C)}, [x_3]_{IND(C)}, [x_4]_{IND(C)}, [x_5]_{IND(C)}, [x_6]_{IND(C)}$ 。决策系统 IS 的改进差别矩阵见表 2。

表 1 决策系统 IS

U	a	b	c	d	e	f	D
x_1	0	0	1	0	1	0	0
x_2	1	1	1	1	1	0	1
x_3	1	0	1	1	0	0	1
x_4	0	0	1	0	0	1	1
x_5	1	0	0	0	0	1	1
x_6	0	0	1	0	1	1	0
x_7	0	0	1	0	1	1	1
x_8	0	0	1	0	1	1	1
x_9	0	0	1	0	0	1	1
x_{10}	1	0	0	0	0	1	1

4 结语

基于差别矩阵求取约简的过程中,简便地实现合取范式到析取范式的等价变换是一大难题。本文设计一种基于差别矩阵的约简树构造方法,首先优化改进差别矩阵得到差别集,然后设计约简树描述约简获取的具体过程,最终所得约简树根节点到叶节点的所有路径即对应决策系统全体约简。与当前同类方法相比,本文所提出方法减少了生成和存储差别矩阵的时空开销,简化了获取决策系统全部约简的过程。

参考文献:

- [1] PAWLAK Z. Rough sets [J]. International Journal of Computer and Information Science, 1982, 11(5): 341–356.
- [2] SKOWRON, RAUSZER C. The discernibility matrices and functions in information systems [C]// Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory. The Netherlands: Kluwer Academic Publishers, 1992: 331–362.
- [3] 刘清. 软件集及软推理[M]. 北京: 科学出版社, 2001: 53–55.
- [4] 庞彦军, 刘开第. 计算约简的差别矩阵简化算法不成立[J]. 系统工程理论与实践, 2004, 23(2): 142–144.

- [1] 唐敏, 姜灵敏, 阳爱民. 一种基于区域模糊特征的图像检索方法 [J]. 郑州大学学报: 自然地理学版, 2007, 39(2): 122–127.
- [2] TANG MIN, YANG AI-MIN, JIANG LING-MIN, et al. A region-based image retrieval method with fuzzy features [C]// International Conference Fuzzy Information Engineer 2007. Berlin: Springer-Verlag, 2007: 940–948.
- [3] RUSSELL S J, NORVIG P N. Artificial intelligence: A modern approach [M]. New Jersey: Prentice Hall, 1995.
- [4] (美) MITCHELL T M. 机器学习 [M]. 曾华军, 张银奎, 译. 北京: 机械工业出版社, 2003: 166–169.