第37卷　第14期　　　　　　　计　算　机　工　程　　　　　　　2011年7月
Vol.37　　No.14　　　　　　　Computer Engineering　　　　　　　July 2011

·软件技术与数据库·　　文章编号：1000—3428(2011)14—0050—03　　文献标识码：A　　中图分类号：TP39

# 基于特征索引的图相似查询过滤算法

邹晓红，李　甲，郭景峰，柴　然

(燕山大学信息科学与工程学院，河北 秦皇岛 066004)

**摘　要：**分析图相似查询候选集的产生过程以及特征图之间的关系对候选图集的影响，提出一种基于特征索引的图相似查询过滤算法，使用 GIndex 算法建立特征图索引结构，通过特征图之间的选择性关系给出一个有序的特征集，并借助特征-图矩阵对数据库进行筛选得到候选图集。实验结果证明，该方法能准确地产生候选图集，从而提高图查询的效率。

**关键词：**相似查询；图数据库；特征索引；特征图；过滤算法

# Graph Similarity Query Filtering Algorithm Based on Feature Index

**ZOU Xiao-hong, LI Jia, GUO Jing-feng, CHAI Ran**

(College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China)

**【Abstract】**This paper analyzes the generation process of the graph similarity search candidates, and the relationship between feature-graph that not be token into account in current filtering algorithm, and presents graph similarity query filtering algorithm based on feature index. It builds a feature-graph index structure, analyzes the selectivity of each feature and builds a sorted feature, and generates the candidate graphs by using the feature-graph matrix to efficiently filter the graph database. Experimental results show that this algorithm can accurately generate candidate and improve the efficiency of the graph query.

**【Key words】**similarity query; graph database; feature index; feature graph; filtering algorithm

**DOI**: 10.3969/j.issn.1000-3428.2011.14.015

## 1　概述

随着信息技术的发展，图作为一种能模拟复杂事物之间联系的数据结构在数据挖掘中越来越重要，图已经在很多领域中被应用，如化学元素、社会网络、模式匹配、XML 等。与图应用密切相关的一个普遍而关键性问题是如何高效查询和检索图数据库，该问题之所以难解决是因为它涉及了图同构的问题，而图同构的问题已被证明是一个 NP 完全问题。因此，目前的图查询策略主要是采用一种图数据库建立索引-过滤-精华提纯的过程。由于精华提纯过程主要是检查图同构的问题，因此目前主要是在前 2 个过程的研究，尽量形成一个较小的候选集。

已经有很多图数据库索引过滤的研究，针对分子结构的 Daylight[1]算法，针对半结构数据和 XML 的 T-Index[2]和 DataGuides[3]算法，这些算法大部分是基于路径和子树的结构。针对更广泛的图模型数据的索引算法有基于路径的 GraphGrep[4]算法，针对基于路径的算法的优点是其对路径的操作比树和图更简单，其索引空间也可以提前定义，其缺点是图数据库中的路径太多形成的集合太大，路径过于简单从而不能包含充足的信息，文献[5]提出基于特征图的 GIndex 算法，特征图是一些比较小的可区分的图块，其包含边之间的关系，又不会产生大量的集合，有效避免了基于路径算法带来的缺点，之后提出 Grafil 算法，解决了图相似查询中的候选图过滤的问题，但是 Grafil 算法未能很好地处理特征图之间的关系，因此本文基于该算法权衡了特征图之间的关系提出基于特征索引的图相似查询过滤算法——SFGF。

## 2　基本概念

**定义 1**(相似查询) 给定图数据库 $D=\{G_1,G_2,L\ ,G_n\}$ 和查询图 $Q$，相似查询是寻找 $D$ 中大体包含查询图 $Q$ 的图。

**定义 2**(放松查询和放松边) 给定图数据库 $D=\{G_1,G_2,L\ ,G_n\}$ 和查询图 $Q$，放松查询是指不查询 $Q$，而是查询 $Q'=Q-E$ 是否被 $D$ 包含，其中，$E$ 为 $Q$ 中任意边的集合，$E$ 中的任意一条边都被称为放松边，$E$ 的大小被称为放松边条数。

**定义 3**(松弛度) 给定 2 个图 $G$ 和 $Q$，$Q$ 为查询图，如果 $P$ 是 $G$ 和 $Q$ 的公共最大公共子图，那么 $G$ 和 $Q$ 的相似程度可定义为$|E(P)|/|E(Q)|$，松弛度为 $1-|E(P)|/|E(Q)|$。

**定义 4**(特征图$f$) 特征图为在图数据中用于构造索引所选取的可以区分数据库中图的一些频繁的小的图结构。

**定义 5**(可区分度) 给定一系列索引图 $F=\{f_1,f_2,\cdots,f_n\}$作为索引，现加入另一图块 $x$，那么 $P(x\,|\,f_{\varphi_1},f_{\varphi_2},L\ ,f_{\varphi_i}),f_{\varphi_i}\subseteq x$，$1\leqslant\varphi_i\leqslant n$，即 $D_x/\mathrm{I}_{f\in F\wedge f\subseteq x}D_f$ 为可区分能力，其倒数 $\mathrm{I}_{f\in F\wedge f\subseteq x}D_f/D_x$ 为 $x$ 对 $F$ 的可区分度。

**定义 6**(DFS(Depth First Search)编码) 使用五元组边对图的一种深度遍历表示方法。

图 1 是本文使用 GIndex 算法建立的索引结构，其构造过程为：首先对数据库中的图使用不同大小的图块使用不同的支持度规则生成频繁图块，再利用图块可区分度得到特征图，最后依据 DFS 编码规则建立索引结构。其中，每个节点都代

**基金项目：**国家自然科学基金资助项目(60673136)；河北省教育厅自然科学研究计划基金资助项目(2009101)

**作者简介：**邹晓红(1967－)，女，副教授、博士研究生，主研方向：数据挖掘，数据库技术；李　甲，硕士研究生；郭景峰，教授、博士生导师；柴　然，硕士研究生

**收稿日期：**2011-02-22　　**E-mail：**lijia-1111@hotmail.com

表一个图，每一层上的图都有相同的大小，第 1 层上的图仅包含一个图节点，以后随着层的增加图增加相应的边。由于无法从某些特征图直接增加边到所需的另一个特征图，因此为了建立索引添加一些过渡图块，如从 $f_1$~$f_3$ 增加过渡图 $f_2$。

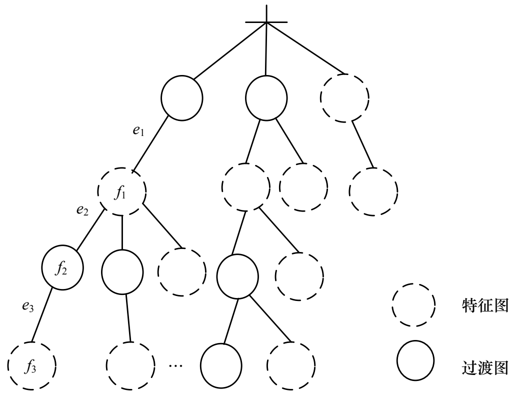



**图 1　使用 GIndex 算法建立的索引结构**

## 3　基于特征索引的图相似查询过滤算法

### 3.1　Grafil 算法思想

Grafil 算法是一种高效的图相似查询过滤算法，它提出图查询包含逻辑对数据库图进行修剪，以下给出包含逻辑[6]，给定查询图 $q$ 与数据库图 $g \in D$，如果特征 $f \subset q$ 且 $f \not\subset g$，则 $q \not\subset g$，即如果特征 $f$ 包含在 $q$ 中，那些不包含 $f$ 的图被修剪。针对 Grafil 只是局限地与图相似查询，对其进行修改，给定查询图和查询图放松边得到的一系列特征图 $F=\{f_1, f_2, \cdots, f_n\}$，然后对比 $F$ 在 $g$ 中出现的次数和在 $q$ 中出现的次数。其实现方法主要是通过把查询图对边的放松查询转化为最大允许的损失特征数，之后利用特征图矩阵对其进行筛选。

**定义 7**(损失特征数)　特征图 $F$ 在查询图 $Q$ 中出现的次数与特征图 $F$ 在放松查询图 $Q$'中出现的次数的差。

表 1 表示特征-图矩阵实例，图矩阵行表示某个特征图 $f_i$，列表示数据库中目标图 $G_j$，每个单元格表示这行对应的特征图包含在此列目标图的次数。

**表 1　特征-图矩阵**

| 特征图 | 目标图 | | |
|---|---|---|---|
| | $G_1$ | $G_2$ | $G_3$ |
| $f_1$ | 0 | 2 | 2 |
| $f_2$ | 0 | 2 | 0 |
| $f_3$ | 2 | 3 | 1 |

Grafil 的过滤方法主要是目标图 $G$ 包含查询图 $Q$，设 $F$ 为特征图的集合，并且 $F$ 满足以下条件，特征图 $F$ 在目标图 $G$ 中出现的次数 $N_g=\sum_{i=1}^{i=|F|}$($f_i$ 在目标图 $G_i$ 中出现的次数)应大于等于 $N_q=\sum_{i=1}^{i=|F|}$($f_i$ 在目标图 $Q$ 中出现的次数)。同理，如果对查询图放松查询，即删除 $Q$ 中的任一条边，$F$ 在目标图中出现的次数 $N_g=\sum_{i=1}^{i=|F|}$($f_i$ 在目标图 $G_i$ 中出现的次数)应大于等于 $N_q=\sum_{i=1}^{i=|F|}$($f_i$ 在目标图 $Q$'中出现的次数)–损失特征数。

例如，图 2 给定查询图 $Q$，图 3 为特征图集合 $F$。图 2 中的 $e_1$、$e_2$、$e_3$ 代表图 3 中不同类的边。假定在数据库中找不到任何与查询图匹配的图，此时通过删除或者重新标记操作可以放松对 $e_1$、$e_2$、$e_3$ 中一个边的查询，因为放松操作可以在 $e_1$、$e_2$、$e_3$ 中的任意一个进行，所以不能确定哪种特征图受到放松的影响。但是不论是哪个边被删除；$f_1$、$f_2$、$f_3$ 都至少在被放松的查询图中出现 3 次。即对比原查询图，被放松的查询图最多损失 $f_1$、$f_2$、$f_3$ 组成的特征图集的最大限度 4 次，因为原查询图出现了 1 次 $f_1$、2 次 $f_2$、3 次 $f_3$。利用这些信息可以删除数据库中那些没有至少包含　3 次特征图集的图。例如针对表 1 为目标图便可删除 $G_1$。然而 Grafil 算法无论是对特征集或者在计算特征损失估计时都没有考虑到特征图之间的关系。这样不仅降低了算法的效率而且也无法过滤掉一些不符合条件的图，本文通过对特征图集之间的关系和顺序，重新对特征集进行分类排序，虽然这个过程需要时间，但是最后过滤精度和整体效率都有所提高。

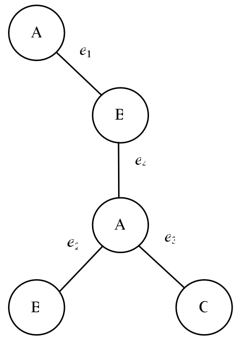
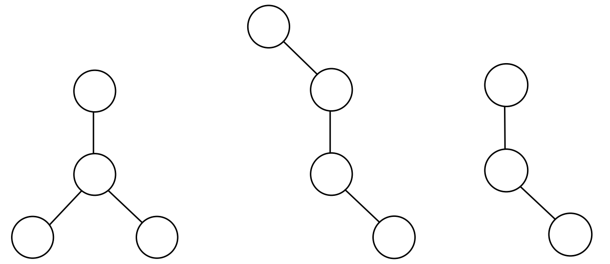



**图 2　查询图 $Q$**



(a)特征图 $f_1$　　(b)特征图 $f_2$　　(c)特征图 $f_3$

**图 3　特征图集合 $F$**

### 3.2　算法改进

Grafil 算法存在的缺点，可通过以下实例进行说明，假如数据库给定一个图 $G$，且图 $G$ 包含样品查询图 $q$，$q$ 为图 2 中 $e_3$ 边被删除的图，$G$ 必须有一次 $f_2$ 内嵌和 2 次 $f_3$ 内嵌。然而可以在图数据库中找到这样一个图，该图不包含 $f_2$ 却包含 3 次 $f_3$，如同表 1 中的 $G_4$，把这种 $f_2$ 的存在少一次而 $f_3$ 却对应增加相应或者更多次数称为特征之间的内联问题。实际上在这种情况下 $G_4$ 是不符合相似要求的。

如果把图 3 中的特征图分成 2 组，则可以解决这个内联问题。让 A 组包含特征图 $f_1$ 和 $f_2$，让 B 组只包含 $f_3$。对于任何图，包含查询图都必须包含 A 组一次，利用这种限制可以过滤图 1 中的 $G_4$。按照这种思想给出 2 个定义并对特征图集设计一个聚合算法。

**定义 8**(特征图 $f$ 的选择度 size($f$))　给定一个图数据库 $D$，一个查询图 $q$ 和一个特征图 $f$，特征图 $f$ 的选择度是指它在 $D$ 和 $q$ 中平均出现频率的差值。

**定义 9**(特征图 $f$ 的顺序)　给定一系列特征集 $F$,$f_1$ 和 $f_2 \in F$，且 $f_1 \neq f_2$，如果 $f_1$ p $f_2$，那么满足如下条件之一：

(1)$select(f_1) < select(f_2)$；

(2)如果 $select(f_1)=select(f_1)$，则 $size(f_1) < size(f_2)$。

根据定义 8 计算出所有特征图的选择度，利用定义 9 对特征图进行排序得到一个有序的特征图集 $F$'，如图 4 所示。

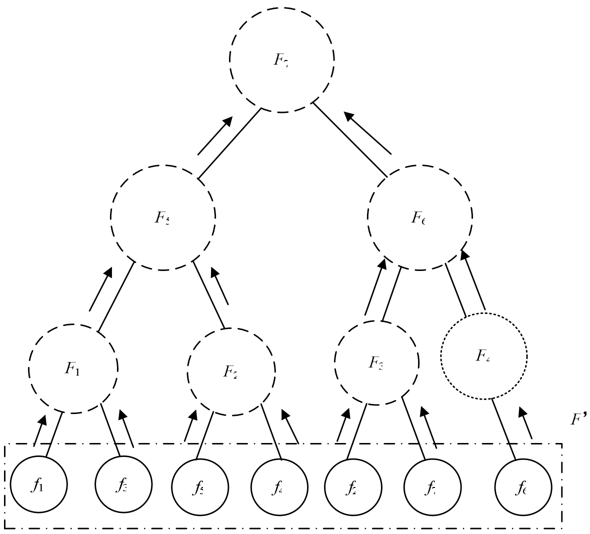



**图 4　特征集的分层结构**

为了把具有相同过滤能力的特征图放在一个单独的过滤组中，把具有相同选择度的特征图放在一个特征集组 $F_i$ 中。最终生成的组呈现出不同过滤能力的特征集。算法从每个特征图都是一个单独的组开始，然后合并 2 个最相近选择度的单独组，由此形成一个特征集的分层树结构，图 4 说明了特征集结构的生成过程。

在合并 2 个组为一个组时，通过平均组选择度和特征图数量 2 个参数把特征图集 $F_i$ 联系起来。被合并的 2 个组的选择度被定义为一个线性插入，新生成的特征集的选择度为：

$$(n_1\delta_1 + n_2\delta_2)/n_1 + n_2$$

其中，$n_1$ 和 $n_2$ 表示组中特征图的数量；$\delta_1$ 和 $\delta_2$ 表示其对应的选择度。

## 4　算法设计和实验

### 4.1　算法设计

**算法 1** SFGF($D$, $F$, max$L$, $Q$)

**输入** 图数据库 $D$，特征图集 $F$，最大特征图大小 max$L$ 和放松查询图 $Q$

**输出** 候选集 $C_Q$

1. 依据特征集建立特征图矩阵
2. $C_Q$←D;
3. F'←重新排序 F;
4. for each f in F';
5. 　计算最大损失特征数 $d_{max}$;
6. 进行组合得到集合{$F_i$};
7. for each $F_i$ in {$F_i$}
8. 　计算此组最大损失特征数 $di_{max}$;
9. 　$C_Q$={G|d(G, Q)≤ $di_{max}$, G∈$C_Q$};
10. for each f in F';
11. 　$C_Q$={G|d(G,Q)≤ $d_{max}$, G∈$C_Q$};
12. return $C_Q$;

**算法 2** GenerateFSet($F$)

**输入** 特征图集 $F$={$f_1$, $f_2$, ⋯, $f_n$}

**输出** 特征图组合集{$F_1$, $F_2$, ⋯, $F_n$}

1. F←重新排序 F
2. while(|F|>1||)
3. 　F=组合新的特征图组合 F'{F'$_1$, F'$_2$, ⋯, F'$_{(n+1)/2}$}
　　其中，F'$_i$=F$_{2i-1}$∪F$_{2i}$，如果 F$_{2i}$=∅，F'$_i$=F$_{2i-1}$
4. 　for each F$_{item}$ in F
5. 　F$_i$ = F$_{item}$
6. 　i++

### 4.2　实验与结果分析

实验环境在一台 CPU 采用 Intel 2.0 GHz Pentium、内存为 1 GB、操作系统为 Windows XP 的机器上完成。算法实现程序由 Visual C++6.0 编译通过。实验数据采用 2 组数据进行验证，一组是化合物结构的 AIDS 数据，这部分数据可以从 http://dtpsearch.ncifcrf.gov/FTP/AIDO99SD.BIN 下载得到，该数据库包含大约 44 000 个化合物，随机选择 AIDS 中 10 000 个图，这些图平均包含 25 个节点和 27 条边，其中，最大的图包含 214 个节点和 217 条边。

这些图中最常见的节点是碳最常见的边式碳-碳边。另一组数据是由 Kuramochi 提供的人工合成数据，人工合成的数据首先随机产生一些种子结构，这些种子结构然后随机组合在一起。

图 5 是使用 AIDS 数据库查询图边为 16 的实验结果，随着放松边数量的增加，得到的筛选候选集合的数量也增加。这个结构和实际相符合。从图 5 可以看出，SFGF 算法和 Grafil 算法的区线轮廓基本一致，同时说明 SFGF 算法的正确性。

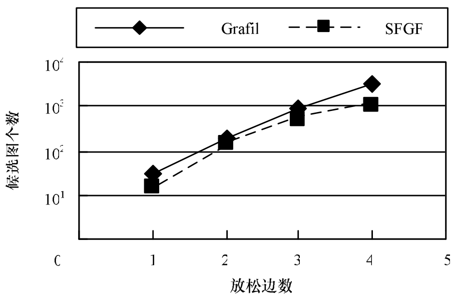



**图 5　基于 AIDS 数据库的查询结果**

图 6 是人工合成数据的实验结果，本文选择一个包含 20 条边的图作为查询图，可以得出，由于使用了分组聚合特征图，因此 SFGF 过滤了那些单纯使用特征图所不能过滤的图，尤其是在查询图放松度较高时所能筛选的结果更高。可见，SFGF 算法的性能优于 Grafil 算法。

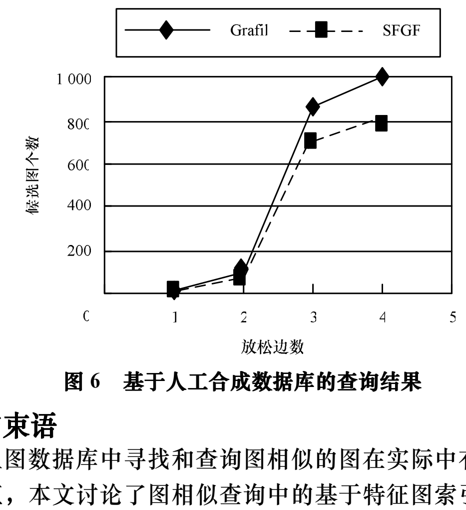



**图 6　基于人工合成数据库的查询结果**

## 5　结束语

从图数据库中寻找和查询图相似的图在实际中有很重要的价值，本文讨论了图相似查询中的基于特征图索引候选集的获取问题，分析了特征图之间的关系对候选图产生的影响，并在此基础上提出一种基于特征图的排序聚类的模型。实验证明，利用该方法能够有效并准确地产生候选图集，从而提高图查询的效率。今后研究的重点为图相似查询过程中的图同构判定。

(下转第 55 页)