1 相关研究

在本章中，我们讨论一些相关的研究。从基本的图数据库索引，子图搜索，模糊子图搜索和图相似性搜索到图核函数

1.1 子图搜索

现在许多的子图搜索都采用了一个相似的框架，先把图分解为一系列小片段，然后将其作为特征，并基于其建立基于特征的索引结构来进行子图查询。属于这类的方法有 *GraphGrep,gIndex,FG-Index,Tree+Delta* 和 *GDIndex*。

在图索引中，最简单的特征就是通路 (walk)。一些情况下，我们可以用路径作为特征，就像这个领域的先驱方法 *GraphGrep*. 路径很好检索特性，并且很好处理。但是，路径的简单性却制约着这种方法的性能。举例而言，即便两图的所有路径都是不同长度的，我们利用路径也不能区分出环和链的拓扑结构。

当意识到路径的局限性后，*gIndex* 和 *FG-Index* 便应运而生了。这两种方法利用子图结构来有效分辨路径和环。但是，基于子图结构的索引也有一些限制，就是子图的枚举和匹配都是运算量非常大的过程。为了克服这些障碍，这些方法采用只提取**频繁子结构**来作为索引特征。还有些相似的方法，像 *Tree+Delta* 和 *TreePi* 则用频繁树结构来代替频繁子图结构来克服这些障碍。

1.2 模糊子图搜索

1.3 图相似性搜索

通常，我们有很多方法去测量图之间的相似度。第一种方法是**编辑距离** *(edit distance)*. **编辑距离**就是我们将图 G 通过一系列操作 (增删点边，重新标号等) 变换为另一个图 G' 所需的操作数。我们可以通过给不同操作分配不同的费用，然后用费用总和当做距离来调整这个方法的准确度。虽然编辑距离是一种非常直观的图相似性测度方法，但是实际上我们难以计算它 (是个 NP-hard 问题).*C-Tree*[**?**] 是一种被广泛使用的图索引模型。它没有使用图的片段信息作为特征值，而是把数据图组织在一种内部节点是**图闭包** *(graph closures)*, 叶节点是数据图的树形结构中。相比于前两种方法 *GraphGrep* 和 *gIndex,C-Tree* 的最大优势在于其支持相似性搜索，而前两个并不支持。

还有一种名为 *GString* 的子图相似性查询方法也和 *GraphGrep* 一样是用图片段作为特征值的。当然，这种方法与前面两种基于特征值的子图查询还是不同的。在这个方法中，首先我们分解复杂图为节点数较少的连通图，得到的这些连通图每个都是一个特定的图片段。随后，我们用一种标准的编号方式把数

据库中的所有图都转化为一个个字符串。并用这些字符串构建一颗后缀树来支持相似性搜索。这种方法融合了子图的数据表达能力 (信息完整) 和用字符串匹配来查询图的速度 (速度快)。

另外, 最大公共子图 *(maximal common subgraph)*[**?**] 和图配对 *(graph alignment)*[**?**, **?**] 这两种方法也常被用来定义图相似度。虽然有这么多方法，但是不幸的是迄今为止我们仍没有一种简单的方法来索引或者度量大图数据库。

1.4 图核函数 Graph Kernel Functions

目前业界有很多图核函数，而开创性的一个图核函数是 Haussler 在他对 *R 卷积核 (R-convolution kernel)* 的研究中提出的。现在大多数图核函数都遵循它提出的这种框架。R 卷积核是基于把离散的结构 (如图) 分解成一系列的组成元素 (子图) 这个思想的。我们可以定义许多这样的分解, 就像组成成分中的核心一样。R 卷积核保证无论选择怎样的分解方式或者成分核心，都能得要一个对称的，半正定的正函数，或者一个成分间的核函数