图数据管理与挖掘

洪 亮 著

科学出版社

图数据管理与挖掘

洪 亮 著

科学出版社

北 京

内 容 简 介

本书介绍了图数据管理与挖掘的关键技术，涵盖基于集合相似度的子图匹配查询处理方法与原型系统、情境感知的个性化推荐方法、利用多层聚簇的跨类协同过滤推荐算法、基于潜在主题的准确性 Web 社区协同推荐方法、基于用户-社区全域关系闭包的高效均衡性 Web 社区推荐方法、Web 社区推荐原型系统、大规模时空图中人类行为模式的实时挖掘方法、基于潜在引用图数据的专利价值评估方法、基于专利关联的新颖专利查找方法，以及异构专利网络中的竞争对手主题预测方法。

本书适合计算机、信息管理等相关专业的高年级本科生和研究生阅读，也可作为数据科学等相关领域的研究与开发人员的参考书。

图书在版编目 (CIP) 数据

图数据管理与挖掘/洪亮著. —北京：科学出版社，2016.11
ISBN 978-7-03-050617-7

Ⅰ. ①图… Ⅱ. ①洪… Ⅲ. ①图象数据处理－研究
Ⅳ. ①TN911.73

中国版本图书馆 CIP 数据核字(2016)第 271845 号

责任编辑：赵艳春 / 责任校对：郭瑞芝
责任印制：张 伟 / 封面设计：迷底书装

科学出版社出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

北京九州迅驰传媒文化有限公司 印刷
科学出版社发行 各地新华书店经销

*

2016 年 11 月第 一 版 开本：720×1 000 1/16
2017 年 1 月第二次印刷 印张：15
字数：300 000

定价：86.00 元

(如有印装质量问题，我社负责调换)

前　言

最近几年，图数据管理与挖掘技术的发展和应用引起了国内外研究者和工业界的极大兴趣。图作为一种常见的数据表示模型，用于建模复杂数据以及数据之间的关联，例如社会网络、语义网、路网、生物网络、专利网络等。图数据库是一种使用图结构建模被存储数据对象的数据库系统。图数据管理的核心问题是图数据库的查询处理，即基于图模型的结构查询，例如子图匹配查询、路径可达性查询、路径距离查询等。虽然从某种角度上来说，图数据库中的查询也可以用 SQL 语言来表达，利用现有 RDBMS 的查询功能来完成，但是这样的查询性能是非常低的。图数据管理研究的关键点是如何设计有效的索引结构和查询算法来快速地回答图数据库中的结构查询问题。图数据挖掘相比于关系数据库的挖掘更强调的是发现与分析数据之间的关联关系。随着大数据时代的到来，数据的关联关系在数据挖掘和分析的过程中越来越受到重视，是商务智能、决策支持、科学研究等领域的核心问题与难点。对于图数据管理与挖掘查询的研究最早可以追溯到 20 世纪 90 年代。最近，由于社会网络数据，专利网络数据，以及语义网数据等领域大数据的大量出现，引起了对于图数据管理与挖掘的新一轮研究热潮。在最近几年的三大国际数据库顶级会议（SIGMOD，VLDB 和 ICDE）上均有图数据管理与挖掘的相关论文，并且数量与比例逐年上升。

社会网络、时空图以及专利网络具有天然的图数据特征，数据之间的复杂关联以及大数据的产生给管理和挖掘这些数据带来了巨大的挑战。本书以图数据理论与模型为基础，面向社会网络、时空图、专利网络等应用领域，提出了一系列的图数据管理与挖掘关键技术。

本书的撰写得到武汉大学多位教师、同学的大力协助和支持，尤其是余骞博士和冯岭博士对本书部分内容的撰写做出了贡献，对他们的辛勤付出表示由衷的感谢！感谢相关学术研究的合作者，你们在我学习和研究道路上给予了大量的帮助和指导。感谢家人的陪伴、支持和鼓励。

本研究受到国家重点研发计划“科学大数据管理系统（面向特定领域的大数据管理系统）”子课题“图数据管理关键技术及系统”（编号：2016YFB1000603），国家自然科学基金青年基金项目“移动社会网络中基于信任关系的情境感知推荐研究”（编号：61303025），国家自然科学基金重大研究计划“大数据驱动的管理与决策研究”重点支持项目“基于知识关联的金融大数据价值分析、发现及协同创造机制”（编号：91646206），以及国家自然科学基金重点国际合作研究项目“大数据环境下的知识组织与服务创新研究”（编号：71420107026）的资助，作者在此表示衷心的感谢。

由于作者水平有限，书中难免存在不妥及疏漏之处，敬请读者批评指正。

目　录

第 1 章　大图数据库中基于集合相似度的子图匹配查询处理方法

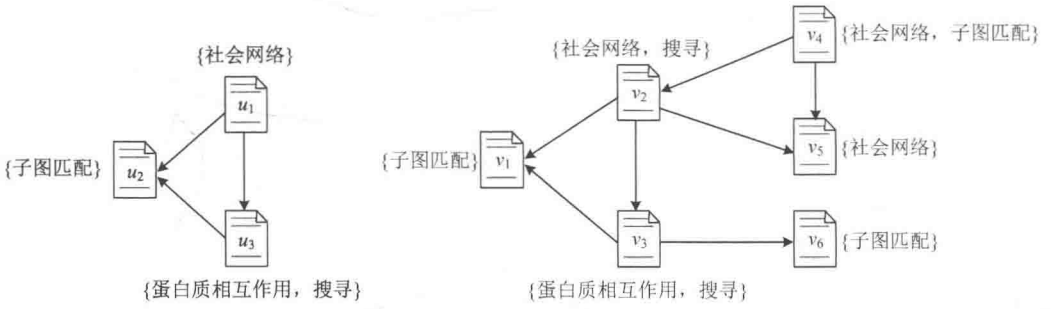
1.1　引　　言

在很多现实应用中，例如社会网络、语义网、生物网络等[1-6]，图数据库作为重要工具已经被广泛用作建模和查询复杂图数据。学者们广泛研究了关于图的各种查询，其中子图匹配[7-10]是一种基本的图查询类型。给出一个查询图 Q 和一个大图 G，一种典型的子图匹配查询是检索 G 中那些在图结构和顶点标签两方面都准确匹配 Q 的子图[7]。然而，在一些图形的应用中，每个顶点往往包含了一系列代表该顶点不同特征属性的元素，而且顶点标签的准确匹配常常是难以实现的。

基于上述内容，本章重点关注一种子图匹配查询的变种，即运用集相似度的子图匹配（SMS^2）查询，其中每个顶点都与一个权重动态变化的元素集合联系起来，而不是一个简单的标签。元素权重根据不同应用环境的要求或包含的不同数据、用户不同的查询要求决定。特别地，给出一个包含 n 个顶点 $u_i(i=1,\cdots,n)$ 的查询图 Q，SMS^2 查询检索大图 G 中所有 n 个顶点 $v_j(j=1,\cdots,n)$ 的子图 X，要求 X 满足：① $S(u_i)$ 和 $S(v_j)$ 间的集相似度大于用户相似度阈值，其中 $S(u_i)$ 和 $S(v_j)$ 分别是与 u_i 和 v_j 相联系的集合；② u_i 和 v_j 对应时 X 与 Q 同构。接下来用两个示例来证明 SMS^2 查询是有用的。

例 1-1：从 DBLP 中找到需要的论文。

DBLP 提供了一个引文图 G 如图 1-1(b)，其中顶点代表论文，边代表论文之间的引证关系。每篇文章包含一个关键词集，其中每个关键词被赋予了一个权重用于测度它在文中的重要性。事实上，一个研究者会基于引证关系和文章内容相似度来从 DBLP 中寻找论文[11]。例如，一名研究人员希望找到同时被社会网络方面和蛋白质相互作用方面的论文引用的子图匹配方面的论文。此外，这名研究人员需要被社会网络方面文章引用的蛋白质相互作用网络研究方面的论文。这样的查询可以建模构成 SMS^2 查询问题，其中包含从 G 中找到匹配查询图 Q（图 1-1(a)）的子图。Q 中的每篇论文即顶点和其在 G 中的匹配论文应该有相似的关键词集，而且每个引证关系即边应当准确符合研究人员的要求。



(a) 查询图 Q　　(b) DBLP 的引证关系图 G

图 1-1　从 DBLP 中找到匹配查询引证图的引证论文的示例

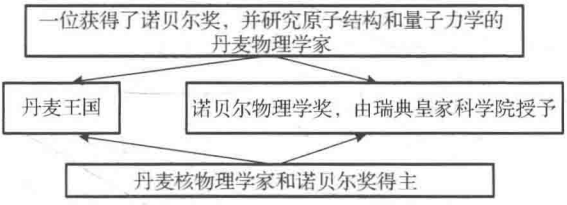
例 1-2：查询 DBpedia。

DBpedia 从维基百科中提取实体并储存在一个 RDF 图中[12]。正如图 1-2(b)所展现的，在一个 DBpedia 的 RDF 图 G 中，每个实体即顶点都有一个属性“dbpedia-owl:abstract”，它提供了一种人工可读的实体的描述，而每条边则是一个表明了实体间关系的事实。特别地，用户提出 SPARQL 查询通过指定准确的查询标准来找出匹配查询图的子图。然而事实上一个用户可能不知道准确的属性值或 RDF 架构（例如属性名称）。例如，一名用户希望找出两个都获得过诺贝尔奖且在 DBpedia 中都与词条“丹麦”相联系的物理学家，同时该用户不知道 DBpedia 的数据架构。这种情况下，该用户可以提出一个 SMS^2 查询 Q，如图 1-2(a)所展现的那样，其中每个顶点都由一小段文字进行描述。这个 SMS^2 查询的结果是 Niels Bohr 和 Aage Bohr，因为该子图匹配与 Q 同构且匹配顶点对代表的文章内容相似度很高。有趣的是，我们发现 Niels Bohr 是 Aage Bohr 的父亲。

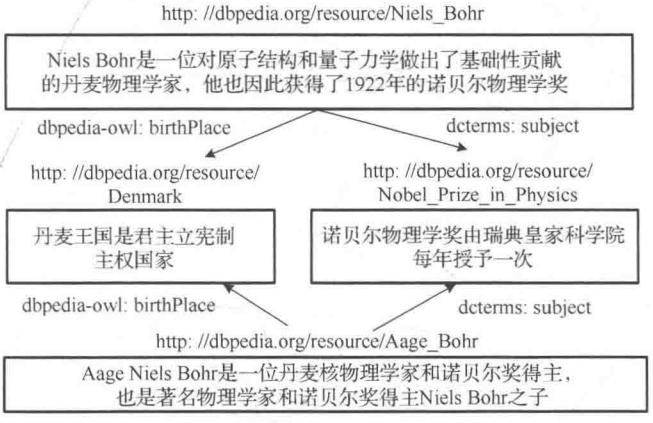
上述两个例子展现了 SMS^2 查询在现实应用中的有效性。当下并没有对同构和运用动态元素权重的集相似度的语义下的子图匹配问题进行研究的工作。传统的加权集相似度[13]关注固定元素权重，这其实是动态权重的集相似度的一种特殊情形。由于顶点的不同匹配语义，之前的精确或近似子图匹配技术[7-9,14-16]并不能直接应用于 SMS^2 查询的问题。

运用动态加权集的相似度和结构约束来有效解决 SMS^2 查询是一个具有挑战性的问题。有两个直接的方法对现有的算法进行修改后解决了 SMS^2 查询问题。第一种方法是运用已有的子图同构算法进行子图同构，如文献[15]和文献[17]所进行的那样，然后通过考察每一对匹配顶点间的加权集相似度对候选子图进行提炼。第二种方法将步骤顺序进行了颠倒，先通过计算加权集相似度在数据图中找出与查询图中的顶点有相似集的候选顶点，这一步在计算复杂度上代价高昂，接下来再得到匹配的子图。然而这两种方法通常会造成高查询损耗，尤其是面对大图数据库。这是因

为第一种方法忽略了加权集相似度的限制条件，而第二种方法在过滤候选结果时忽略了结构信息。



(a) 不考虑DBpedia数据架构的查询图Q



(b) DBpedia的RDF图G

图 1-2　在 DBpedia 的 RDF 图中得到查询匹配子图的示例

由于已有的方法都不高效，本章提出一种有效的 SMS^2 查询实施方法。本章的方法采取一种过滤加提炼的框架，对图拓扑和动态加权集相似度的特点加以利用。在过滤过程中，为数据图 G 建立一个关于其顶点元素集的频繁模式的基于格的索引。然后将数据顶点进行编码表示，并组织进一个签名桶。在基于格的索引和签名桶的基础上，设计一种有效的剪枝技术来减小 SMS^2 搜索空间。在提炼过程中，提出一种基于支配集（Dominating Set，DS）的子图匹配算法来找出运用集相似度的匹配子图。一种支配集选择方法被提出用于选择一个经济的查询图的支配集。综上，本章主要有以下几点贡献。

（1）设计了一种新颖的有效解决 SMS^2 查询的技术。一个倒排的基于格的索引和一种结构化的基于签名的局部灵敏散列（Locality Sensitive Hashing，LSH）都在线下过程中首先被施行。在线上过程中，一系列剪枝技术将被应用和整合，从而更好地减小 SMS^2 查询的搜索空间。

（2）提出了一种运用了创新的关于数据顶点的元素集的倒排模式格的集相似度

剪枝技术，用来度量动态加权的集相似度。该方法引入一种动态加权的相似度的上界来运用反单调原则，以得到更好的剪枝效果。

（3）提出了一种结构化的剪枝技术，探索出一种新颖的基于结构化标记的数据组织结构，其中标记被设计用于捕捉集合与相邻集的信息。一种聚集支配（aggregate dominance）原则被设计出来用于引导剪枝过程。

（4）不同于直接的查询和校验查询图中的所有候选顶点，设计了一种有效的算法来完成基于查询图的支配集展开子图匹配过程。当补充完剩余顶点后，距离保留原则被设计出来以裁剪掉那些不能与支配顶点保持距离的候选顶点。

（5）最后通过实验证实了本章方法可以有效解决大图数据库中的 SMS^2 查询问题。

1.2 预 备 知 识

1.2.1 问题定义

本小节将正式定义基于集合相似度的子图匹配查询问题。特别地，考虑一个大图 G，将其表示为 $\langle V(G),E(G)\rangle$，其中 $V(G)$ 代表顶点集，$E(G)$ 代表边集。每个顶点 $v\in V(G)$ 都与一个元素集合 $S(v)$ 相对应。查询图 Q 被表示为 $\langle V(Q),E(Q)\rangle$。支配元素集记作 u，其中每个元素 a 都有一个权重 $W(a)$ 来表示其重要性。注意到由于实际应用中的不同需求和数据，这个权重是可以在不同的查询中动态变化的。

定义 1-1：基于集合相似度的子图匹配。对一个含有 n 个顶点 $\{u_1,\cdots,u_n\}$ 的查询图 Q，一个数据图 G，以及一个用户特定的相似度阈值 τ，Q 的一个子图匹配将是 G 的一个包含 $V(G)$ 的 n 个顶点 $\{v_1,\cdots,v_n\}$ 的子图 X，如果满足以下条件：

（1）存在函数 f，使得所有 $V(Q)$ 中的顶点 u_i 和 $v_j\in V(G)(1\leqslant i\leqslant n,1\leqslant j\leqslant n)$，满足 $f(u_i)=v_j$；

（2）$\mathrm{sim}\big(S(u_i),S(v_j)\big)\geqslant\tau$，其中 $S(u_i)$ 和 $S(v_j)$ 都是分别与 u_i 和 v_j 联系的集合，且 $\mathrm{sim}\big(S(u_i),S(v_j)\big)$ 得到一个 $S(u_i)$ 和 $S(v_j)$ 的集合相似度值；

（3）对所有边 $(u_i,u_k)\in E(Q)$，存在 $\big(f(u_i),f(u_k)\big)\in E(G)(1\leqslant k\leqslant n)$。

由于基于集合相似度的子图匹配是独立于直接或间接的边的，因此可以用于直接或间接的图中。接下来定义 SMS^2 查询。

定义 1-2：SMS^2 查询。给出一个查询图 Q 和一个数据图 G，运用集合相似度的子图匹配查询检索在集合相似度语义下 G 中 Q 的所有子图匹配。

注意到相似度函数 sim 的选择将高度依赖于应用范围。本章选择加权的雅加达相似，这是一种广泛用于计算相似度的方法。

定义 1-3：加权的雅加达相似。给出关于顶点的元素集，则加权的雅加达相似表示为

$$\mathrm{sim}\big(S(u),S(v)\big)=\frac{\sum_{a\in S(u)\cap S(v)}W(a)}{\sum_{a\in S(u)\cup S(v)}W(a)} \tag{1-1}$$

其中，$W(a)$表示元素 a 的权重。

在定义 1-3 中，当所有元素 a 的权重都为 1 时，加权的雅加达相似则变成经典雅加达相似[18]。正如文献[18]中所描述的，其他一些相似度计算方法如余弦相似、哈明距离、重叠相似度等都可以转换成雅加达相似。因此给出任意一种其他的相似度度量方法和阈值，都可以转化为雅加达相似并与阈值进行比较。然后运用一个常量下界来实施 SMS^2 查询。最后利用原始相似度度量来校验每一个候选点。

在实际应用中，每个元素 a 的权重可以被查询提出方或加权工具例如 TF/IDF[13]等确定。例如在例 1-1 中，每个关键词的权重代表着关键词与论文间的联系，这将由研究人员来确定；在例 1-2 中，DBpedia 中的每个词条可以被赋予 TF/IDF 权重，这将根据涉及的数据来动态变化。

1.2.2　架构

本小节将给出一个过滤与提炼的方法架构，其中包括线下过程与线上过程，如图 1-3 所示。

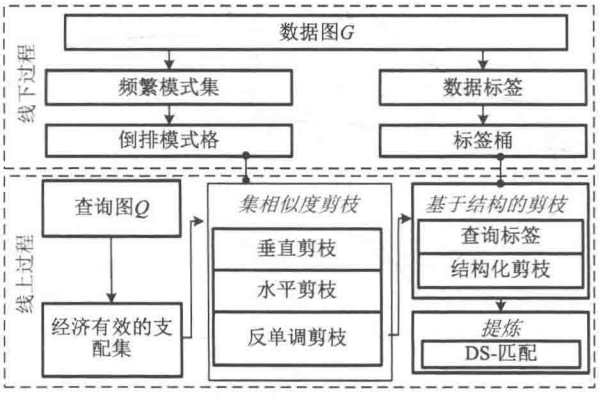


图 1-3　SMS^2 查询过程架构

在线下过程中，构建一个创新的倒排模式格用于更好地执行基于集相似度的有效剪枝。由于元素的动态权重使得已有的索引对解决 SMS^2 查询问题无效，需要为 SMS^2 查询设计一种新的索引。受格结构的反单调特征启发，从数据图 G 的顶点元素集中挖掘频繁模式。如果频繁模式 P 包含于顶点元素集中，则为该频繁模式 P 储

存数据顶点到倒排列表中，格和倒排序列一起成为倒排模式格，它能很好地减少动态加权的集相似度搜寻导致的损耗。为了支持结构化的剪枝过程，通过考虑集合和拓扑信息将每个查询和数据的顶点编码表示为查询签名和数据签名，并混合所有的签名进入签名桶。

在线上过程中，找出一种关于查询图 Q 的经济的支配集，且只搜索支配集中的候选顶点。注意到不同的支配集将导致不同的查询结果，因此提出一种支配集选择算法来选择一种经济的支配集 DS(Q)。

基于支配集的子图匹配过程将基于以下两点展开：①在 SMS^2 查询中找出候选点将比在子图搜索中耗费更多，因为集相似度计算比顶点标签匹配更不经济。因此，过滤处理的损耗能够通过搜索支配顶点而不是所有查询顶点来减小。②一些查询顶点可能有很多候选顶点，这将在子图匹配过程中产生很多不必要的中间结果，因此子图匹配损耗也可以通过降低中间结果的规模得到减少。

对 DS(Q)中的每个顶点 u，提出一种剪枝策略，包括集相似度剪枝和基于结构的剪枝。集相似度剪枝包括反单调剪枝、垂直剪枝、横向剪枝，这些剪枝过程都是建立在倒排模式格上的。基于签名桶，提出一种运用顶点签名的基于结构的剪枝策略。

剪枝过程之后，基于 DS(Q)中的候选支配顶点，提出一种有效的 DS-匹配的子图匹配算法来得到 Q 的子图匹配。DS-匹配运用了控制顶点和非控制顶点间的拓扑关系来减小子图匹配过程中的中间结果，并因此减小匹配损耗。

1.3　集合相似度剪枝

对查询图 Q 的支配集中的顶点 u，需要找出其在图 G 中的候选顶点。根据定义 1-1 中对 SMS^2 查询的定义，如果图 G 中的顶点 v 与 u 对应，则有 $\mathrm{sim}(S(u),S(v))>\tau$。本小节将详述如何找出满足条件的候选顶点 v，而如何选择经济的支配集将在后文中进行介绍。

已有的索引都依赖于元素规范化，因而不适用于 SMS^2 查询，尽管如此，注意到两个集合间的包含关系即使在元素权重发生变化时也不会改变。对分别对应顶点 v 和 v' 的两个元素集合 $S(v)$ 和 $S(v')$，如果 $S(v)\subseteq S(v')$，则称前者为后者子集的关系为包含关系。基于包含关系，引出如下的上界。

定义 1-4：AS 上界。给出一个查询顶点 u 的对应集合 $S(u)$ 和数据顶点 v 的对应集合 $S(v)$，一种反单调相似度（AS）上界为

$$\mathrm{UB}(S(u),S(v))=\frac{\sum_{a\in S(u)}W(a)}{\sum_{a\in S(u)\cup S(v)}W(a)}\geqslant \mathrm{sim}(S(u),S(v)) \tag{1-2}$$

其中 $W(a)$ 表示元素 a 被赋予的权重，由公式（1-1）给出。

由于当查询给出后 $\sum_{a\in S(u)} W(a)$ 就不会改变，AS 上界是关于 $S(v)$ 反单调的，即对任意的 $S(v)\subseteq S(v')$，如果 $\mathrm{UB}\big(S(u),S(v)\big)<\tau$，则 $\mathrm{UB}\big(S(u),S(v')\big)<\tau$。

显然 AS 上界的反单调性质使我们能够基于包含关系来进行顶点的剪枝。然而数据顶点的元素集间的包含关系数量很少，相反，由于大部分元素集包含频繁模式，元素集和频繁模式间的包含关系数量很多。因此，本章从所有数据顶点的元素集中挖掘频繁模式，并设计了创新的索引结构倒排模式格，用于组织频繁模式。基于格的索引可用于实施有效的反单调剪枝，并因此实施集相似度搜索。这里借用频繁模式的定义[19]。

定义 1-5：频繁模式。用 u 表示 $V(G)$ 中的突出元素，模式 P 是 u 中一系列元素的集合，即为 u 的子集。如果元素集 $S(v)$ 包含所有 P 中的元素，则可以说 $S(v)$ 支持 P 且是 P 的一个支持元素集。P 的支持表示为 $\mathrm{supp}(P)$，指支持 P 的元素集的数量。如果 $\mathrm{supp}(P)$ 比用户确定的阈值更大，则 P 被称为频繁模式。

模式 P 的支持 $\mathrm{supp}(P)$ 的计算方式在文献[19]中有介绍。

1.3.1　倒排模式格的构建

为了建立一个倒排模式格，首先需要从数据图 G 的所有顶点元素集中挖掘频繁模式，然后将所有频繁模式组织为一个格。在格中，每个连接点 P 是一个频繁模式，它是其自身所有下位点的子集。将有 k 个元素的频繁模式表示为 k-频繁模式。为了确定索引的复杂度，1-频繁模式被归入格的第一层。对格中每个 k-频繁模式 P，将每个顶点 v 的对应元素集 $S(v)$ 插入 P 的倒排序列 $L(P)$中，当且仅当 $S(v)$ 支持 P。注意到一个元素集 $S(v)$ 可能支持不同的频繁模式，将其分别插入这些频繁模式的倒排序列中。

例 1-3：图 1-4 是一个倒排模式格的示例，它由图 1-1(b)中的数据顶点建立。元素 a_1、a_2、a_3、a_4 分别对应关键词子图匹配、蛋白质相互作用、社会网络、搜寻。然后 $v_1=\{a_1\}$，$v_2=\{a_3,a_4\}$，$v_3=\{a_2,a_4\}$，$v_4=\{a_1,a_3\}$，$v_5=\{a_3\}$，$v_6=\{a_1\}$。

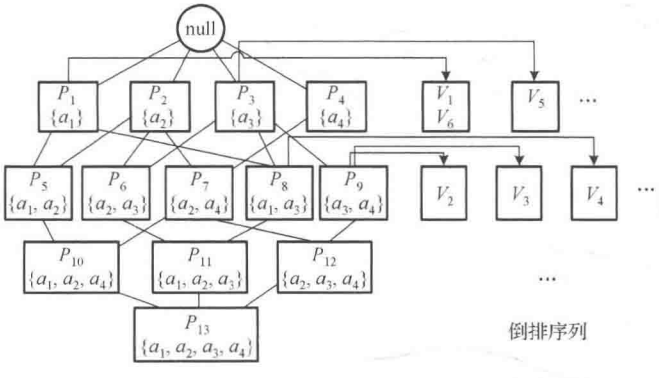


图 1-4　包含频繁模式格和倒排序列的倒排模式格