# 一种有效率的基于图的关系学习算法*)

郑丽珍[1] 郭景峰[2] 李 晶[1] 边伟峰[1]

(燕山大学信息科学与工程学院 秦皇岛 066004)[1] (河北工业大学电气与自动化学院 天津 300130)[2]

**摘 要** 多关系数据挖掘根据表示形式可以分为基于图的 MRDM 和基于逻辑的 MRDM。本文讨论了基于图的数据挖掘和基于图的关系学习之间的关系,重点介绍基于图的关系学习算法 Subdue 及其优缺点,针对它的缺点提出优化的算法 ESubdue,改进了子图同构的计算,减少了子图同构的次数。在实际和人工数据集上运行的实验结果显示它比原算法更加有效率。最后给出结论并指明将来的工作。

**关键词** 多关系数据挖掘,基于逻辑的 MRDM,基于图的 MRDM,Subdue

## An Efficient Graph-based Relational Learning Algorithm

ZHENG Li-Zhen[1] GUO Jing-Feng[2] LI Jing[1] BIAN Wei-Feng[1]

(Department of Information and Engineering, Yanshan University, Qinhuangdao 066004)[1]

(Department of Electricity and Automation, Hebei University of Technology, Tianjin 300130)[2]

**Abstract** Multi-relational data mining can be categorized into graph-based and logic-based according to their representation. We talk about the relationship between graph-based data mining and graph-based relational learning. An overview on different methods for graph-based data mining is given. We mainly discuss graph-based relational learning algorithm Subdue, including its advantage and disadvantage. To solves the disadvantages of Subdue, we propose ESubdue, which improve the subgraph isomorphism computation and reduces the times for subgraph isomorphism. Experimental results on both real and synthetic datasets indicate that the improved algorithm is much more efficient than the original one. Finally we conclude the paper and talk about the future work.

**Keywords** Multi-relational data mining, Logic-based MRDM, Graph-based MRDM, Subdue

## 1 前言

近年来,数据挖掘在理论和实际应用上都有了迅速发展。早期数据形式局限于单个关系表或事务,每个实例由表中的一行或一个事务表示。随着应用领域的扩展,出现了对结构数据挖掘的研究。最近多关系数据挖掘(Multi-relational data mining, MRDM)成为与结构数据挖掘相关的研究热点,它致力于从复杂的、多关系的和结构的数据中发现涉及多个关系的复杂模式。在发现过程中,数据的表示是一个根本且关键的问题。MRDM 一般使用两种形式的表示:基于图的表示和基于逻辑的表示[1]。

基于逻辑的 MRDM,也就是归纳逻辑编程(Inductive Logic Programming, ILP)。它的特点是使用逻辑表示多关系数据,实例、背景知识、假设和目标概念都使用 Horn 子句逻辑表示。ILP 系统中的 FOIL[2]、CProgol[3]、TILDE[4] 和 WARMR[5]等已经成功地应用于 MRDM 的很多领域。ILP 的核心是使用逻辑表示,从由背景知识提供的谓词中搜索句法有效的假设。它擅长发现语义复杂的概念,并且能有效地利用背景知识。

ILP 系统要在数据预处理阶段把关系数据库中的数据转换为逻辑程序的形式,在数据挖掘阶段,算法在逻辑程序上运行。这种方式在现实使用过程中有诸多限制,包括输入格式过于严格、缺乏效率、缺乏对关系数据库性质的考虑以及无力处理数据中的噪声和缺失值等[6]。

基于图的 MRDM,一般称为基于图的关系学习。它以图的形式表示多关系数据,例子、背景知识、假设和目标概念都表示成图。基于图的 MRDM 系统已经广泛地应用于无监督学习(一般称为频繁子图挖掘)和有监督学习。这些方法包括基于图论的方法,如 FSG[7] 和 gSpan[8];基于贪心搜索的方法,如 Subdue[9] 和 GBI[10];内核函数的方法[11]也已经用于基于图的关系学习。这些方法的核心是使用基于图的表示,搜索频繁的或压缩输入图的或能够区分正例和负例的图模式,它擅长发现大型的结构概念。

## 2 相关工作

基于图的数据挖掘(Graph-based data mining, GBDM)和基于图的关系学习(Graph-based relational learning, GBRL)有很强的联系。基于图的关系学习可以看作是基于图的数据挖掘的一个分支,因为基于图的数据挖掘的任务是找到有效地挖掘嵌入在图中的频繁模式的方法,而基于图的关系学习的主要任务是找到新颖的、有用的、可理解的图模式,而这样的模式不仅仅是频繁的。但是基于图的数据挖掘的很多方法可以应用于基于图的关系学习中。

基于图的数据挖掘已经成功地应用于各种领域,包括蛋白质分析、化合物分析、链接挖掘和 Web 搜索。从图数据集中挖掘有趣的图模式已经有很多成功的技术和方法。这些方法包括基于图论的方法、基于贪心搜索的方法、基于内核函数的方法等。

*)国家自然科学基金项目(编号:60673136)。**郑丽珍** 硕士生,主要研究方向为多关系数据挖掘、分类等。

基于图论的方法主要使用支持度或频率度量来挖掘一个完整的子图集。这个领域最初是由基于 Apriori 思想提出的 AGM[12]算法研究，它利用邻接矩阵每次增加一个顶点来逐层增加图的大小，得到所有的(连通的和不连通的)诱导子图。FSG 使用类似的方法，在运行时间上进行优化，利用邻接矩阵每次增加一条边来逐层增加图的大小，得到所有连通的频繁子图。gSpan 使用 DFS 编码来规范标记，采用深度优先搜索策略，在内存和运行时间上比以前的方法更加有效率。

基于图论的图挖掘方法的一个共同特征，是它们挖掘全部的频繁子图。与之对比的是 Subdue 系统，它使用最小描述长度(minimum description length, MDL)的原则评定一个子结构的有趣度。使用 MDL 而不是频率评定一个子结构是 Subdue 和其它图挖掘算法的主要不同。它一般产生较少数量的、能够最好地压缩图数据集的子结构，这些子结构能够提供关于这个领域的更加有用的知识。下一节将详细介绍 Subdue 的工作原理和优缺点。

## 3 Subdue 算法介绍

### 3.1 相关定义

**定义 1** 标记图 一个标记图 $G$ 表示为一个四元组 $(V(G), E(G), L_V, L_E)$、顶点集 $V(G)$、边集 $E(G)\subseteq V(G)\times V(G)$、一个结点标记集 $L_V$、一个边标记集 $L_E$。

结点的数目 $|V(G)|$ 称为图的大小。

**定义 2** 子图(subgraph) 图 $H$ 是图 $G$ 的子图，它满足 $V(H)\subseteq V(G)$，$E(G)\subseteq E(H)$，且 $H$ 中边的端点的分配和 $G$ 中的一样，用 $H\subseteq G$ 来表示“$G$ 包含 $H$”。

**定义 3** 连通图 设 $u$ 和 $v$ 是图 $G$ 的两个不同的顶点，若 $u$ 和 $v$ 之间存在一条路径，则称 $G$ 为连通图，否则称为不连通的图。

对于大多数应用，连通子图就足够了，所以本文只考虑连通子图的挖掘。

**定义 4** 子图同构(subgraph isomorphism) 从图 $G$ 到图 $H$ 的同构是一个双射 $f: V(H)\to V(G)$ 使得 $uv\in E(G)$ 当且仅当 $f(u)f(v)\in E(H)$。如果存在从 $G$ 到 $H$ 的同构，称“$G$ 同构于 $H$”。标记图的子图同构还要求保持对应结点和边的标记一致。

**定义 5** 子结构(substructure) 一个子结构是输入图中一个连通的子图。

**定义 6** 实例(instance) 一个子结构的实例是输入图中与子结构同构的子结构。

### 3.2 Subdue 算法

Subdue 是一个基于图的关系学习系统，它既可以用于有监督学习，又可以用于无监督学习。数据表示成标记图，每个结点表示对象或属性，每条边表示对象之间的关系，输入可以是一个图也可以是一个图集，输出是根据 MDL 原则能够最好地压缩输入数据库的子结构。它采用横梁搜索算法，最初把每一个结点标记作为一种子结构，通过增加一条边或增加一条边和一个结点来扩展一个结点，产生候选子结构。Subdue 保存在图例中子结构的实例，使用图的同构来决定在图例中候选子结构的实例。子结构然后根据它们对数据集的描述长度的压缩程度来评估。重复这个程序直到所有的子结构都考虑过或者超过用户指定的要扩展的子结构的数目。最后程序返回，最好压缩子结构。

Subdue 算法作为多关系学习算法具有如下优点：大多数基于图的数据挖掘方法的一个共同特征是它们挖掘全部的频繁子图，像 FSG 和 gSpan 保证挖掘全部的满足用户指定约束的频繁子图。虽然完整性是一个令人期望的性质，但是不能忽视这些系统产生大量子结构却提供相对很少的领域知识。通常，有趣的子结构还需要领域专家或其它自动方法来识别，从而获得某个领域的有用知识。Subdue 一般产生较少数量的能最好压缩图数据集的子结构，这些子结构能够提供关于这个领域的更重要的知识。

Subdue 算法虽然具有上面所述的优点，但实验证明 Subdue 算法在效率上远远低于 FSG 和 gSpan。

## 4 优化的 Subdue 算法

针对 Subdue 算法运行效率不高的问题，本文提出优化的 Subdue 算法 ESubdue。

基于图的数据挖掘的核心问题是如何避免冗余搜索。因为按照不同的顺序增加相同的结点，相同的子图会以不同的方式产生，很难保证每个子图只考虑一次。虽然同样子图进行多次测试不影响结果，但这极大地增加了算法的执行时间。对冗余子图的核查需要子图同构的算法，因此能够排除这种冗余搜索的方法对算法的效率很重要。

Subdue 算法运行效率不高的主要原因是没有采用特定的搜索策略来减少冗余子图的产生，产生冗余子图后又没有有效率的子图同构算法。子图同构是一个 N-P 问题，提高算法的效率，一定要具备有效率的子图同构算法。

规范形式的核心思想是建立一个编码来唯一确定一个图，使子图同构变得有效率。这个编码的特点是描述图的边，特别是与结点相连的边。如果是标记图或有向图，它们也包括标记或边的方向，还有附属结点的标记。下面用无向图的例子来介绍规范标记，有向图可以转化成无向图来处理，参见文[12]。

图结构一般使用邻接矩阵来表示，定义一个图的规范标记最简单的方式是使用通过图的邻接矩阵来获得。图的编码通过将邻接矩阵的上三角的每列依次连接起来获得。因为不同的顶点排列对应不同的邻接矩阵，所以有不同的编码。把其中最大的编码称为规范编码，它可以唯一地表示一个图。图 1 是图规范编码的一个例子，图 1(a)是一个大小为 3 的图 $G$ 和它的两个不同的邻接矩阵。其中图 1(c)的编码是“$aaazyx$”，其中“$aaa$”是把结点标记按照它们出现在邻接矩阵中的顺序连接起来，“$zyx$”是把邻接矩阵的上三角的每列依次连接起来获得。图 1(c)的编码是所有编码中最大的，由它获得的编码叫作规范编码，对应的邻接矩阵叫作规范邻接矩阵。顶点的另一种排列会有不同的邻接矩阵和编码，但其它排列产生的编码都比图 1(c)的小。图 1(b)的编码是“$aaazxy$”小于“$aaazyx$”。

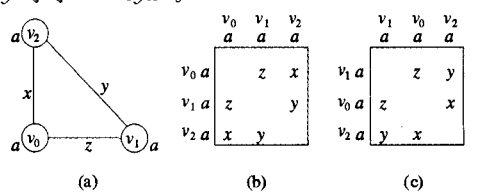


图 1 图和规范邻接矩阵的例子

如果一个图有 $|V|$ 个结点，决定一个图的规范标记的时间复杂度是 $O(|V|!)$。在 FSG 中利用结点的不变性将结点分割为不同的等价类，在每个类内部之间排列，还有各种优化

来减少获得规范编码的时间。详细内容参见文[7]。

Subdue 算法中把每一种结点标记都作为一种子结构，这样做的效率也不高。我们采用先扫描一遍图数据集，将数量小于某个用户指定的值 MinLim 的标记去掉，这对有很多不频繁标记的数据集来说会明显减少一阶候选集合 $C_1$ 的大小，从而减少后面判断子图同构的次数，这也会提高 Subdue 的运行时间。算法 1 描述了改进后的算法 ESubdue。

**算法 1** ESubdue(GD, BeamWidth, Limit, MinLim)

```
输入:GD:图数据集;BeamWidth:横梁宽度;
     Limit:用户指定的计算约束;
     MinLim:标记出现次数的最小值;
输出:发现的最好子结构
①初始化 ParentList, ChildList, BestList = Null; ProcessedSubs =
  0;
②扫描一次 GD, 为每一种标记的出现次数计数;
③if (标记出现次数≥ MinLim )
④为这种标记和它的实例建立子结构,将子结构插入 ParentList;
⑤while (ProcessedSubs<= Limit and ParentList not empty)
⑥  do
⑦    while (ParentList is not empty)
⑧     do
⑨       Parent= RemoveHead (ParentList);
⑩       以所有方式扩展 Parent 中的每个实例;
⑪       将扩展的子结构分组到 Child 结构中;
⑫    for each Child
⑬       转化为规范编码来评价 Child;
⑭       将 Child in ChildList in order by value;
⑮       if BeamWidth <=Length (ChildList)
⑯       then destroy the substructure;
⑰  ProcessedSubs++;
⑱return discovered best substructures;
```

## 5 实验

在实际的和人工合成的数据集上，比较 ESubdue 和 Subdue，以及基于图的数据挖掘算法 FSG 和 gSpan。实验环境为 Intel Pentium 1.5GHz 的 PC 机，内存为 512 MB，操作系统为 Windows 2000 server。

### 5.1 实际的数据集

Mutagenesis 数据集是基于分子结构来识别化合物的突变性，它被认为是多关系数据挖掘的基准数据集。Mutagenesis 数据集由 230 个化合物的分子结构组成，化合物的突变性已经由 Ames 试验(检查致癌物质的测试)确定，其中 138 个标记为突变的，92 个标记为非突变的。在这个数据集上的任务是找到基于化合物的分子结构特征来区分突变的和非突变的化合物。Mutagenesis 数据集基本上由原子、键、原子类型、键的类型和原子的局部电荷组成。

分别运行 ESubdue 和 Subdue 在 138 个正例，两个算法发现相同的子结构。另外，把 ESubdue 和 Subdue、FSG、gSpan 在该数据集上的运行时间进行对比。把 FSG 和 gSpan 的结果在 10%的支持度产生 1956 个子结构，如果支持度小于这个值，会产生更多随机和不重要的模式。表 1 显示了 FSG，gSpan，Subdue 和 ESubdue 在该数据集上的运行时间。

表 1 在 Mutagenesis 数据集上的运行时间

| 算法 | 时间(秒) |
|---|---|
| Subdue | 142 |
| FSG | 21 |
| gSpan | 4 |
| ESubdue | 36 |

在实际的数据集上显示 Subdue 和 ESubdue，能够发现相同的子结构。因为 ESubdue 第一遍扫描数据集，去掉的是不可能是最优子结构一部分的结点，不会影响最后挖掘的结果，但减少了后面产生冗余子图的次数，从而提高了运行效率。更加有效的子图同构也是算法效率提高的原因。虽然 ESubdue 比 Subdue 运行时间提高了很多，但是算法的运行时间高于 FSG 和 gSpan。我们可以采用更完善的子图产生过程来进一步减少冗余子图的产生，这也是我们将来的工作之一。

### 5.2 人工数据集

为了验证 ESubdue 的运行时间是否随着数据集的大小线性增长，我们采用 Subdue 提供的图数据集产生器，分别产生包含 500，1000，1500 和 2000 个事务的图数据集。

表 2 在不同大小的数据集上的运行时间

| 事务的数目 | FSG | gSpan | Subdue | ESubdue |
|---|---|---|---|---|
| 500 | 734 | 61 | 51 | 48 |
| 1000 | 815 | 107 | 139 | 102 |
| 1500 | 882 | 182 | 169 | 154 |
| 2000 | 1112 | 249 | 696 | 314 |

表 2 中实验结果显示 Subdue 的运行时间不是线性的，尤其是当数据集的数量很大时运行时间急剧增长，ESubdue 比 Subdue 的运行时间明显减少，ESubdue 的运行时间基本上是随着数据集的大小线性增长。但是，ESubdue 的运行时间仍高于 FSG 和 gSpan，我们可以采用更完善的子图产生过程来进一步减少冗余子图的产生，这也是我们将来的工作之一。

**结论** 针对基于图的数据挖掘系统 Subdue 的效率不高的问题，我们提出了 Subdue 的优化算法 ESubdue，它包括一个更好的子图同构算法，减少了子图同构计算的次数，提高了算法的效率，使得算法执行时间随着数据集的大小线性地增长。但是 ESubdue 没有有效的子图扩展策略来减少冗余子图的产生，所以运行时间不是最优的。另外，现在横梁的宽度是基于整个数据集的大小而定。有效的子图扩展策略和决定最优的横梁宽度是将来的研究方向。

## 参 考 文 献


1 Ketkar N S, Holder L B, Cook D J. Comparison of graph-based and logic-based multi-relational data mining. SIGKDD Explorations, 2005, 7(2)

2 Quinlan J R. Learning logical definitions from relations. Machine Learning, 1990, 5:239～266

3 Muggleton S. Inverse entailment and progol. New Generation Compute, 1995, 13(3-4):245～286

4 Blockeel H, Raedt L D. Top-down induction of first-order logical decision trees. Artif. Intell., 1998, 101(1-2):285～297

5 Dehaspe L, Toivonen H. Discovery of frequent datalog patterns. Data Min. Knowl. Discov., 1999, 3(1):7～36

6 Ngo L, Haddawy P. Answering queries form context sensitive probabilistic knowledge bases. Theoretical Computer Science, 1997, 171(1-2): 147～177

7 Kuramochi M, Karypis G. An efficient algorithm for discovering frequent subgraphs. IEEE Trans Knowl Data Eng, 2004, 16(9): 1038～1051

8 Yan X, Han J. gSpan: Graph-based substructure pattern mining. In ICDM, 2002. 721～724

9 Cook D J, Holder L B. Graph-based Data Mining. IEEE Trans on Intelligent Systems, 2000, 15(2):32～41

10 Matsuda T, Horiuchi T, Motoda H, et al. Extension of graph-based induction for general graph structured data. In: PAKDD, 2000. 420～431

11 Kondor R I, Lafferty J D. Diffusion kernels on graphs and other discrete input spaces. In: ICML, 2002. 315～322

12 Inokuchi A, Washio T, Motoda H. An apriori based algorithm for mining frequent substructures from graph data. In: PKDD, 2000. 13～23