

基于三维空间的概念格自动布局^{*})

马骏¹ 沈夏炯^{1,2} 刘宗田²

(河南大学计算机与信息工程学院 河南开封 475000)¹

(上海大学计算机工程与科学学院 上海 200072)²

摘要 论文讨论了几种常见的概念格自动布局形式,通过对各种自动布局方式特点的分析,提出了概念格的三维自动布局策略,对三维空间中的层次布局与层内节点自动布局问题进行探讨并加以实现,较好地解决了二维布局中格节点横向过度扩张和线段交叉的问题。

关键词 概念格,形式概念分析,格的可视化,自动布局

Automated Layout of Concept Lattices Based on Three-dimensional Space

MA Jun¹ SHEN Xia-Jiong^{1,2} LIU Zong-Tian²

(Computer and Information Engineering College, Henan University, Kaifeng 475000)¹

(Computer Engineering and Science College, Shanghai University, Shanghai 200072)²

Abstract This paper suggests a three-dimension policy for automatic layout of concept lattices based on analyzing several relative methods, and implements the solution by researching the key topics including layers layout of lattice node in three-dimensional space and nodes layout in a layer. Thereby, this policy solves the problems in a plane such as horizontal over-expanding of nodes, and line segments overlapping.

Keywords Concept lattice, Formal concept analysis, Lattice visualization, Automatic layout

1 引言

概念格是形式概念分析中核心的数据结构,通过形式化地描述对象和特征之间的联系,揭示了概念之间的泛化与例化关系。概念格的可视化给人们提供了直观的分析与观察知识单元内在联系的方法,然而,概念格的布局却是相当困难的。在概念格的图形表示中,有两种主要的布局手段,一种是手工布局,另一种是利用计算机自动布局。手工布局的优点是能够根据人们的审美观去修改每个节点的位置,从而得到一种高质量的概念格图形。然而手工布局费时费力,效率不高。因此如何利用计算机进行概念格的自动布局自然受到了人们的高度重视。

自动布局的最大困难在于如何尽可能“美观”地显示任意给定的概念格,如何既能“美观”地显示概念格又能方便地实现人机交互操作,成为一个新的热门研究课题。目前,比较典型的有附加布局^[2]、分层布局^[4]以及力定位布局^[5,6]等,然而,它们共同的特点是多数都采用了二维构图的策略,对于较大的概念格,节点横向过度扩张和线段交叉仍是一个无法有效解决的问题。由于概念格的结构决定了它的空间伸展特性,因此本文提出三维布局的策略。经过实现,较好地解决了二维布局中的上述问题,另外,三维布局可以更加准确地描述越来越复杂的格结构,以及方便地实现互操作,应该是知识可视化更加合适的解决方案。

本文首先介绍了几种流行的概念格的布局形式及其特点,然后提出了基于三维空间的布局策略并加以实现,最后给

出了工作总结和进一步研究的内容。

2 概念格的可视化形式

概念格的可视化就是用表格或者图形的方式直观地描述概念格。由于图形描述可以清晰地表达形式概念及其相互之间的关系,给人带来一目了然的感觉,因此用图形描述概念格就变得非常重要。Wille 在文[1]中这样解释:格不仅限于一种数学描述,而且还蕴含了很多意义。因此,将格用图形表示出来不仅仅是反映一种数学结构,而是给出了一种有多种意义的数据表示形式或表示方法。

2.1 线图

线图(Line Diagram)是最基本的概念格表示方法。在线图中,每一个节点表示一个形式概念,然后把符合 Hasse 图性质的关节点用连线连接起来,从而使概念之间的结构变得清晰和易于理解。

线图是以 Hasse 图为基础的,在线图中,每一个节点都是一个形式概念,在这种表示法中,将节点 X_1 画在节点 X_2 的下面表示 $X_1 \leq X_2$ 。为了表示图中每一个节点的含义,可以给节点加上标签,即每一个节点用上下两个标签表示,上面的标签表示概念的内涵,下面表示概念的外延,如果两个节点存在偏序关系,就用一条连线将它们连在一起。

连线图使得给定数据背景的概念结构变得清晰和易于理解,为实现概念格的其它自动布局方式奠定了基础。

2.2 附加线图

附加线图(additive line diagram)是由 Ganter 和 Wille 引

^{*})本文受国家自然科学基金(60275022)、河南省自然科学基金(0311011700)和上海市高等学校青年发展基金(03AQ99)资助。马骏 硕士,副教授,研究方向是并行计算、知识发现及网络应用。沈夏炯 博士研究生,副教授,研究方向是软件工程、知识发现、分布式/并行计算及分布式存储。刘宗田 博士生导师,教授,主要研究领域为人工智能和软件工程。

人的^[2],Ganter 和 Wille 是这样描述附加线图的:设 L 是一个格, rep 是格 L 中的节点表达函数,对于格 L 引入表达集 P , P 中的每一个元素都是一个二维向量,记为 $vec(p)$,函数 $rep: L \rightarrow \varphi(P)$ 表示对格中的每一个元素,都有 P 中的一个向量与之对应,约束条件为:

$$\forall x_1, x_2 \in L: x_1 \leq x_2 \Leftrightarrow rep(x_2) \subseteq rep(x_1)$$

其中 x_1, x_2 是概念格中的两个概念。

为了保证 Hasse 图的特征,规定表达集 P 中的每一个元素都用正的纵向分量表示,而横向分量则不作限制,于是格中每个概念 a 的坐标位置可以定义为:

$$pos(a) = n + \sum_{p \in rep(a)} vec(p)$$

其中 n 是一个向量,用于调整坐标系中的偏移量,其值可以根据图的要求任意选择。

有两种表示附加线图的方法,一种是属性表示法,另一种是属性-对象表示法。在属性表示法中,直接把属性集作为表达集,即定义表达集为:

$$rep(x) = Int(x)$$

其中 $Int(x)$ 表示概念 x 的内涵。

属性表示法的优点是:如果要找某个概念的内涵属性有哪些,可以分解该位置的向量,所得到的分向量就是其内涵。缺点是有些概念的纵向位置可能在图中扩张得太厉害,由于在基于属性表示法的图形中,纵向的分向量都是正值,因此如果一个具有大量属性的概念的内涵与它的父节点的内涵不同的话(比如图中最下面的节点),就会导致该节点与其父节点的纵向距离可能比所有节点的平均距离要大得多。

为了解决这个问题,可以考虑将对象也作为表达集中的成员,对于概念格 $\underline{B}(G, M, I)$ 中的概念 (A, B) ,定义概念 (A, B) 的表达集是其内涵 B 和不包含其外延 A 的格中其它所有对象的并集,即:

$$rep((A, B)) := B \cup (G \setminus A)$$

其中 $G \setminus A$ 表示从对象集 G 中去掉子集 A 所得到的新的对象集。

这样,就可以得到概念格的另一种表示形式,通过这种方法得到附加线图的方法称为属性-对象表示法。

用属性-对象表示法表示的附加线图虽然解决了某些节点扩张太厉害而导致图形在纵向层次上脱节的问题,但是,该方法仅适用于比较简单的特定概念格,对一些复杂的概念格仍然无法得到理想的效果图。

2.3 分层图

附加线图虽然是自动绘制概念格常用的方法之一,但是使用该方法构造的图形一般无法满足美观性要求,而将节点分层布局^[4]则是另一种概念格绘制方法。这种画法既保留了附加线图的结构,又达到了美观的一般性要求。

概括地说,分层图可以通过三个步骤完成:

- 1) 首先确定每一个顶点所在的层;
- 2) 层与层之间尽量减少边的交叉;
- 3) 每一层中的各顶点在水平方向平均分配。

分层图一般有两种类型,一种是完全由计算机自动绘制,另一种是手工辅助绘制。自动绘制主要通过链分解(chain-decomposition)来实现^[5],该方法非常适用于可分配概念格。在人工绘制中,主要使用几何方法和在附加线图的基础上直接进行变换。

2.4 有向力定位布局

有向力定位布局(Force directed placement, PDP)是布局

图中非常著名的一种技术^[5,6]。它将弹簧原理(距离越大,拉力越大)和电磁力原理(距离越小,斥力越大)应用于布局图中,并寻求一种最小的受力配置。

在弹簧沿边作用于顶点的斥力算法中,力的大小均用下式转换为一个变化速率函数:

$$\frac{\partial \phi}{\partial d_{ij}} = F_{ij}(d_{11}, \dots, d_{n2})$$

此处 F_{ij} 是第 i 个顶点的第 j 个分向量,然后通过移动顶点或者向量进行优化处理。

事实上,一个“好”图或者“美观”的图依赖于视觉上的主观判断,并没有明确的判断标准。然而一个明显的必要条件是尽量避免线段的交叉,在二维图中这个问题更为突出。

3 基于三维空间的概念格自动布局

前面介绍的几种自动布局的一个共同特点是,它们都采用了二维成像的方法,有向力定位布局虽然采用了三维力定位算法,但最终还是映射到二维图中显示出来。虽然二维显示能与最终的输出设备相对应,但是不可避免会产生线段交叉的问题。由于概念格的结构决定了它的空间伸展特性,因此,三维可视化和互操作应该是概念格可视化更加合适的解决方案。概念格的三维可视化策略有如下优点:

1. 每层格节点的布局空间较大,从而避免二维图中可能出现的宽度过大的问题;
2. 观察角度很多,结合旋转等交互操作手段,使得观察者能够更主动地选择视点,从而解决二维图中线段交叉的问题;
3. 便于实现格的“折叠”和“扩展”等子格嵌套操作,使得格结构更加清晰。

在具体实现中,我们首先使用组件技术构造三维概念格图形,然后通过设置组件相应的属性或者调用相应的方法实现概念格图形的显示与处理功能,例如调整概念节点大小、显示与隐藏节点标签、只显示部分偏序关系、整体缩放、旋转与变换、调整间距与观察点、读入格结构、手工调整节点位置等。

3.1 算法描述

布局图中主要涉及到两个布局算法,一个是分层算法,用于决定概念格节点所在的纵向坐标位置;第二个是平面节点分配算法,用于决定同一水平面中节点的坐标位置。

3.1.1 分层算法

在三维布局中,概念格中的每一个元素仍然通过分层函数 $layer$ 分配一个唯一的层号,层与层之间的垂直距离平均分配。下面介绍与分层函数相关的定义。

定义 1 设 a 是概念格中的一个形式概念,则 a 的分层函数定义为从 a 开始到格的上确界(即最顶层节点)的最大路径上边的个数之和。

显然, $layer$ 满足以下性质:

性质 1 设 L 是一个格, $K_1 = (G, M, I)$ 是一个形式背景, a, b 为概念格 $\underline{B}(K_1)$ 中的两个形式概念, $layer(a), layer(b)$ 为两个概念所在的层号,如果对于任意的 $a, b \in \underline{B}(K_1)$, 都满足 $a < b \Leftrightarrow layer(a) > layer(b)$ 。

从定义 1 可以看出,分层函数是以形式概念的内涵为基础的(越往下其属性集中的元素个数越多),即对任意的形式概念 a 和 b , 如果 a 比 b 小(即 a 的属性个数比 b 的属性个数多),则根据分层函数得到的布局图中,形式概念 a 肯定在形式概念 b 的下面。

在具体实现中,首先遍历对应数据库表中每个形式概念,

并计算出每个概念的所在的层号,然后将其保存到二维数组中。二维数组的第一列保存概念的名称,第二列保存概念所在的层号。

3.1.2 平面节点分配算法 对于同一层中的节点,为了得到较好的质量,可以根据实际情况决定同一层中的所有节点是线性分布或者是沿水平面平均分布。这样就可以使三维布局既能适用于简单的概念格图形,也能适用于复杂的概念格图形。

在同一层中,可以采用下列几种布局方法之一:

◆ **线性分配法:**将同一层中的节点在水平面按线性平均分配。

◆ **层次分配法:**将同一层中的节点在水平面按照属性个数重新分层。

◆ **受力定位法:**给同一层中的每一个节点所属的每一个属性分配一个合适的向量,然后利用某种算法确定各个节点的受力大小,当达到受力平衡时,即可确定节点的位置。

◆ **圆形分配法:**将同一层中的节点在水平面按圆形平均分配。

在线性分配法中,可以将节点沿水平方向平均分配,这种方法的优点是简单直观,缺点是同一层节点过多时会导致横向过度扩张。

层次分配法则是将同一层节点重新按照内涵中的属性个数再次在水平面进行分层,算法的主要思想是:首先计算出同一层中每个节点的属性个数,并按照属性个数将节点顺序排列;然后将排列后的节点根据属性个数从中间向两边在水平面重新分层,属性个数相同的节点分配在同一层。对于同一层节点比较少少的情况,这种方法非常类似于线性分配法,当节点比较多时,层次分配法能有效地减少横向扩张,但是该方法的视觉效果不如圆形分配法。

受力定位法对于某些概念格能够得到非常好的视觉效果图,但是由于该方法给节点属性分配合适的向量需要采用很多次试验的方法才能得到,因此只适用于某些特定的概念格,对于任意的概念格该方法效率不高。

圆形分配法则是将节点在水平面平均分布到圆周上,算法分为两种情况:当节点个数小于3时,按线性分配法进行分配,否则按圆形平均分配。这种分配法的好处是充分利用了三维技术,使之既适用于节点较少的情况,也适用于节点较多的场合。

3.2 软件实现

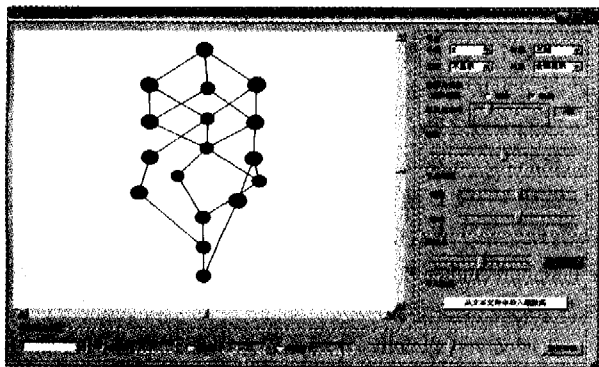


图1 三维布局的部分显示界面

在软件实现中,首先采用组件技术根据已知概念格自动生成三维布局图,并设置相应的属性、方法提供对该组件的操

作,然后由用户根据情况进行各种变换以及调整每个节点在其所在层中的布局,并随时将用户调整的结果保存下来。当所有节点的位置确定以后,再将概念格所有布局信息保存到文件中,以便于重新显示该图。这种方法的好处是将自动绘图和人工绘图相结合,从而大大提高了工作效率和图的质量。

图1,2是三维布局的部分界面,其中图1是从正面观察看到的效果,图2是顺时针旋转一定角度后得到的效果。显然,使用基于三维空间的自动布局更便于观察格的结构,也更加便于用户的互操作。

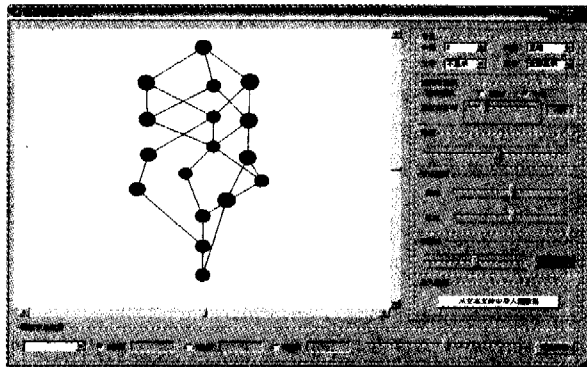


图2 旋转一定角度后的显示界面

结论和进一步的工作 基于三维布局的概念格图形主要解决二维布局中的横向过度扩张和线段交叉的问题,由于三维布局减少了二维布局中的节点调整次数,因此生成概念格图形的速度非常快。鉴于对于简单的概念格,同一层中的布局算法使用线性布局效果更好一些,因此我们在软件实现中分别提供了同一层中的线性布局和平面布局两种功能。随着概念格越来越复杂,三维布局的优越性也越来越明显,但是相应地也增加了人工调整的难度,所以,寻求一种更好的优化算法是今后进一步研究的目标。

概念格的可视化作为知识可视化的一种手段,以其直观的独特优势正在赢得越来越多的研究者关注。进一步的研究内容包括:概念格图形的优化结构描述、自动优化节点变换、子概念格的嵌套折叠与展开、格的变形及布局优化算法等。

参考文献

- 1 Wille R. Lattices in data analysis; How to draw them with a computer. Algorithms and Order, 1993, 33~58
- 2 Ganter B, Wille R. Formal Concept Analysis; Mathematical Foundations. Springer Verlag, 1999
- 3 Luksch P, Skorsksy M, Wille R. On drawing concept lattices with a computer. In W. Gaul and M. Schader, editors, Classification as a tool of research, North Holland, Amsterdam, 1986. 269~274
- 4 Cole R. Automated Layout of Concept Lattice Using Layer Diagrams and Additive Diagrams, Griffith University, Australia, ACSC 2000
- 5 Cole R. Automated Layout of Concept Lattices Using Force Directed Placement and Genetic Algorithms. School of Information Technology Griffith University, Australia, ACSC 2000
- 6 Freese R. Automated Lattice Drawing. Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia