软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(5):1074−1090 [doi: 10.13328/j.cnki.jos.004957]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

树比较可视化方法综述*

李彦龙[1], 李国强[1], 董笑菊[2]

[1](上海交通大学 软件学院,上海 200240)

[2](上海交通大学 计算机科学与工程系,上海 200240)

通讯作者: 董笑菊, E-mail: xjdong@sjtu.edu.cn

摘 要: 层次数据是生产和生活中常见的一类数据.树可视化通常用于呈现层次数据,是可视分析中的一个重要部分.比较是一种常用的分析方式,而树比较可视化也是可视分析研究中的一个重要课题.树比较可视化从所比较的树的数目角度,分为树内比较、两树比较和多树(3 树及以上)比较;从数据特点角度分为静态树比较和动态树比较;从比较分析的任务角度,分为结构比较和属性比较.对现有的树比较可视化方法进行了总结,按照不同方法的表现形式,将树比较可视化分为并置、合并以及动画这 3 类.同时,结合其他分类方法对树比较可视化方法进行了全面的分析和评估.通过评估和比较,分析了不同方法的优缺点以及在不同的数据分析需求中的适用性.在此基础上,还对树比较可视化中的交互方式进行了总结,并阐述了树比较可视化所面临的挑战.

关键词: 可视分析;树比较;层次数据

中图法分类号: TP391

中文引用格式: 李彦龙,李国强,董笑菊.树比较可视化方法综述.软件学报,2016,27(5):1074−1090. http://www.jos.org.cn/1000-9825/4957.htm

英文引用格式: Li YL, Li GQ, Dong XJ. Survey on visualization of tree comparison. Ruan Jian Xue Bao/Journal of Software, 2016,27(5):1074−1090 (in Chinese). http://www.jos.org.cn/1000-9825/4957.htm

Survey on Visualization of Tree Comparison

LI Yan-Long[1], LI Guo-Qiang[1], DONG Xiao-Ju[2]

[1](School of Software, Shanghai Jiaotong University, Shanghai 200240, China)

[2](Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: Hierarchical data is very common in daily life, and tree visualization, which is used to represent hierarchical data, is an important part of visual analysis. While comparative analysis is widely used in problem solving, tree comparison plays an important role in visual analysis. From the perspective of comparing tree numbers, there are inter-tree comparison, pair-wise comparison and multi-tree comparison. Taking data characteristics into consideration, static tree comparison and dynamic tree comparison can be applied. Tree comparison can also be classified into structure comparison and attribute comparison according to comparison tasks. This paper provides an overview of the tree comparison methods relating to visualization area. These methods are divided into three categories according to their representation: juxtaposition, superposition and animation. The paper analyzes both advantages and disadvantages of different approaches, and then gives suggestions in different situations. It concludes with a discussion about the interactive explorations and the challenges of tree comparison.

Key words: visual analysis; tree comparison; hierarchical data

* 基金项目: 国家自然科学基金(61100053, 61472240, 91318301)
Foundation item: National Natural Science Foundation of China (61100053, 61472240, 91318301)
收稿时间: 2015-07-28; 修改时间: 2015-09-19, 2015-11-09; 采用时间: 2015-12-05

© 中国科学院软件研究所 http://www.jos.org.cn

1 引 言

数据可视化是一个跨学科的领域,它将数据信息和知识转化为视觉形式,能够充分发掘人在视觉感知和模式识别方面的优势,帮助人们发现和理解大规模的复杂数据中潜在的规律和特征.数据可视化为人们提供了一种快速进行信息传递和知识发现的方式和手段.在数据分析领域,可视化既是数据分析的关键技术也是数据分析结果呈现的关键技术[1].

树是对现实事物关系的一种抽象,树所代表的层次数据在生活和工作中极其常见,比如生物进化树、系统树图、文件组织结构、公司人员管理关系、家谱图等.Shneiderman[2]根据信息的特征把信息可视化分为一维信息(1-dimensional)、二维信息(2-dimensional)、三维信息(3-dimensional)、多维信息(multidimensional)、层次信息(hierarchical)、网络信息(network)、时序信息(temporal)可视化.树是层次信息的直接形式,树可视化作为信息可视化中的一个重要分类,广泛应用在生产生活以及科学研究中.

“比较”在可视分析中不是一个单一的任务,相反地,它需要用户在一组相关对象上进行一系列操作从而完成一组任务,比如寻找相似、差异分析等[3].由于树特殊的结构特点,树比较可视化在各个领域都有广泛的应用.根据数据源的不同,树比较可视化有 4 个主要的应用领域:生物信息学(bioinformatics)、层次分类(hierarchical classification)或分面分类(faceted classifications)、模式映射(schema mapping)以及计算机软件开发领域[4].

在生物信息学中,对一组生物进行物种关系分析,可以生成一组不同的模型,用系统发生树(phylogenies)和分类树(taxonomies)表示.每个模型都能反映这组生物在生物进化或者生物特性上的相似性.通过比较来分析模型的差异或者对模型进行整合修改,能够增加生物学家对物种的了解,从而建立一个更优的模型[5−10].

对数据进行层次分类或者分面分类时,通常选择一组能够表征数据特性的属性,根据属性值逐层分类,进而形成一个树结构,树的每个叶节点代表一类数据,从根到叶节点的路径代表了这个类别在每个属性上的值的归属情况.而用于分类的属性不同或者使用的属性顺序不同,生成的树结构会有较大的差异,选取合适的属性和顺序对于用户理解数据有很大的帮助.文献[11−14]中使用的美国联邦预算数据、肺癌发病数据等,都是采用选取一组属性对数据逐层划分的方式生成树结构,再进行可视分析.一些数据属性本身就能代表层次信息,比如在 NBA 球员数据[13]中,球员球队所在的分区(东部/西部)、赛区以及城市等属性,在地域上形成了一个层次结构,将球员的数据放在层次结构中比较,能够清晰地表现球员在地域上的分布特点.

对层次异构数据的比较是树比较可视化的直接应用,其关键在于寻找数据中节点的最优匹配.树比较可视化一方面有助于找到最优匹配,另一方面也是一种最直观的结果展示方法.

在软件分析以及源代码分析中,代码的包结构以及代码的语法树都是树结构,所以代码可视化多以树的形式呈现.Beck 等人[15]用树比较的方法帮助用户分析软件结构;Telea 以及 Wettel 等人[16−18]将树比较应用在版本管理系统中,分析代码演变过程和进行错误追踪[19];Leschke 等人[20]用树比较的方法分析目录结构变化,以此来协助数字取证.

除此之外,在涉及层次数据的可视化工作中,大多有比较的任务参与其中.而不同的数据语义和比较任务需要不同的比较方法.研究树比较可视化有助于合理地进行方法选择,因此,本文对树比较可视化方法进行文献综述,以期为树比较可视化方法选择提供参考和依据.

1.1 树可视化

树可视化方法根据节点关系的表现形式分为 4 类:缩进形式、点线链接形式、邻接形式和空间填充式.

使用缩进的形式表示树结构时,如图 1(a)所示,同一级的节点有相同的缩进,每个子节点比其父节点有多一级的缩进.缩进形式的树可视化方法常应用在文件浏览器中,使用缩进来表现文件的目录结构,用户使用折叠和展开操作,可以快速地对文件结构进行浏览并定位文件的位置.缩进形式的缺点在于可扩展性不高,文件目录较多时,一次只能显示一部分,给文件定位带来困难.在实际的应用中,可通过添加搜索功能解决定位困难的问题.

点线链接形式是最基本的树的表示形式,它是用连线表示节点父子关系的一种可视化手段,如图 1(b)所示.在点线链接形式中,每个点表示一个实体对象,实体对象之间的关系用它们之间的连线表示.点线链接形式的优

© 中国科学院软件研究所 http://www.jos.org.cn

点在于它能够清晰地表达数据的层次结构;缺点在于当数据量增大时,节点数量随着深度的增加呈指数级增长,接近根节点的位置节点较少,而远离根节点的位置节点较多,使得空间利用率不高并且空间分配不均匀,远离根节点的位置容易形成聚集.气泡图[21,22]采用的方式是将父节点放在一个圆形的中间,其子节点环绕在周围,递归采用相同的方式布局子节点,直到叶子节点.气泡图在一定程度上提高了空间利用率.其他还有放射布局、双曲线布局以及雷达布局等方式,都通过改变子树延伸的方向来弥补普通从上到下或从左到右布局空间利用率低的缺点.

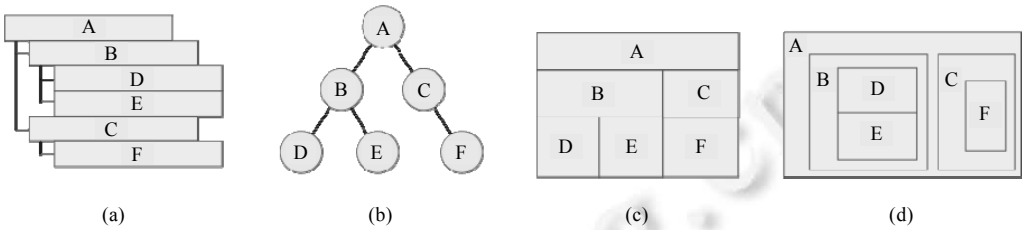


Fig.1 Basic representation of tree visualization

图 1 树可视化的基本表示方法

冰柱图(icicle plot,如图 1(c)所示)是使用邻接关系表示节点父子关系的方式.子节点在父节点的一侧与父节点相邻,所有节点的排列沿某个方向延伸,而叶子节点深度的不同,使得节点排列产生锯齿的形状,很像冬天屋檐下的冰柱,冰柱图因此得名.该方式牺牲了一定的结构性,但是提高了空间利用率;缺点在于:当子树的深度差异比较大的时候,在不同分支下会形成很大的锯齿,同样存在空间利用不均的问题.

树图(treemap)[23](如图 1(d)所示)是空间填充形式的代表,它用图形的包含关系来表示树节点的父子关系,能够将空间利用率最大化.不足之处在于不能清晰地表示树的层次结构,虽然可以通过矩形边框特殊化的方式来增强层次信息,但是会浪费一定的空间,当层次结构复杂的时候效果也不理想.Cushion 树图[24]通过添加阴影效果的方式来强调层次信息.如果数据层次较深且数据较多,低层的元素可能会退化成接近像素点矩形,不容易辨认.Squarified 树图[25]具有良好的纵横比,有序树图(ordered treemap)[26]能够保持节点的布局顺序同数据的存储顺序的一致性,避免局部数据改变时其他节点的位置偏移过大而失去节点的上下文.

Jürgensmann 等人[27]使用可视化的方式对树可视化方法进行了总结.肖卫东等人[28]总结了层次信息可视化技术,并提出了一个基于表现-维度的分类框架.TreeVis.net[29]从维度(dimensionality)、表现方式(representation)以及对齐方式(alignment)这 3 个方面对树可视化文献建立了索引.Shneiderman[30]阐述了树图的种类以及发展.张昕等人[31]对不同的树图布局算法从纵横比、稳定性、可读性等几个方面进行了比较.Schulz 等人[32]从维度、点表示形式、边表示形式以及布局方式这 4 个方面讨论了层次数据可视化方法的设计空间(design space).表 1 中列出了不同的树可视化方法的优缺点.

Table 1 Comparison of tree visualization methods

表 1 树可视化方法比较

形式	方法/布局	优点	缺点
缩进形式	类的加权方法数	方法简便且易于实现,可以使用纯文本	数据量大时需要滚动操作,容易失去上下文,空间利用率低
点线链接	上下/左右布局、径向(radial)布局、气泡图、H-Tree、正交布局、Cone 树	能够直观地表达数据的结构信息	节点数量随深度增加呈几何级数增长(可以使用双曲布局、变形、节点过滤等方式解决)
邻接形式	Icicle Plot, Sunburst	相对于点线链接形式,省略了边的绘制,提高了空间利用率	单一方向上增长过快,树内部结构差异较大时锯齿状明显,空间利用不均衡
空间填充	Slide-and-Dice 布局、Squarified 布局、Ordered 树图、Voronoi 树图、Contrast 树图	空间利用率高,父节点属性值和子节点属性值之间有加和关系	对层次结构的表达不明显

© 中国科学院软件研究所 http://www.jos.org.cn

1.2 树比较可视化

树比较可视化是树可视化中的一个重要内容,从被比较的树的数目角度,可以分为树内比较、两树比较和(三树及)多树比较,在这 3 类中,以两树比较最为常见;从数据特点角度可以分为静态树比较和动态树比较;从比较的任务角度可以分为结构比较和属性比较,数据不同,任务的侧重也不同.完成可视比较的任务需要恰当的方法支持[3],根据其展示方式主要分为 3 类:(1) 并置(juxtaposition);(2) 合并(superposition);(3) 动画(animation).前两种方法主要侧重于用树可视化的方法展示数据,通过空间位置、颜色形状映射等方式展现每个实体对象的关联以及属性值差异;动画的方式则多用于动态树的比较.

根据不同的需求和数据特点,树的比较可以划分为不同的任务类型.2003 年的信息可视化大赛以树的比较为主题,将比较任务分为拓扑结构比较和属性比较.在拓扑比较中,关注树的整体差异、局部差异,节点的增加、删除以及子树之间的差异.树是数据的抽象,树的每个节点一般对应数据中的一个实体对象,属性的比较主要是比较实体对象的属性值变化.比如在某个属性或者某几个属性上,树内部或树之间是不是有明显的差异,值是增加还是减少等等.具有不同特性的数据对应的比较任务的侧重点也不尽相同.

在前人的一些工作中,对可视化中的比较做出了分析和总结:Gleicher 等人[3]从比较方法的设计和交互等方面对信息可视化中的比较进行了总结;而 Pagendarm 等人[33]将比较的形式引入到科学可视化,分析了数据层面和图像层面的比较在科学可视化中的应用;Graham 和 Kennedy 等人[4]对多树可视化进行了文献综述,他们从可视化的树的数量角度出发,分析了对单棵树、两棵树以及多棵树的不同的可视化方法.本文在这些工作的基础上,从展示方式入手,结合其他分类方法对树比较可视化方法进行了总结.

本文第 2 节从表现方式的角度详述并置、合并和动画的树比较方法.第 3 节介绍树比较可视化中的交互.第 4 节讨论树比较可视化中所面临的挑战以及发展.最后,第 5 节对全文进行总结.

2 树比较可视化方法

树比较可视化方法按照其布局方式分为并置、合并、动画这 3 种.表 2 列出了本文涉及到的主要文献.

Table 2 Classified papers

表 2 文献分类

方法		主要文献
并置	直接连接	[11,16,18,34]
	视觉映射	[7,8,10,35,36]
	矩阵	[15,37–41]
合并	直接比较	[9,12–14,17,19,20,42–44]
	显式编码	[14]
动画	–	[13,45,46]

2.1 并 置

并置的比较方法是指使用一般的树可视化方法将比较的数据直接可视化放在同一个视图中,比较通过观察对比的方式完成.并置是最直接的方式,实现上也比较容易,当数据量增多时,方法上也不需要做太多的修改,易于扩展[3].其优点在于,它能够引导用户关注比较的树中的对应节点,适合在比较的树结构差异比较大的情况下比较其中的相似部分,所以并置常用于结构比较.并置的缺点在于,当比较的树结构相似度很高的时候,相同的部分会在每个树结构中都绘制一次,这些冗余导致空间利用率低下,同时增加了寻找对应节点的难度.

直接依赖用户的观察比较会给用户造成记忆负担,用户需要在并置对象上的比较,因此用并置的方法进行树比较的时候,通常会为用户提供视觉暗示(visual cues)帮助理解数据的关联[47].因此,并置的比较方式的核心是依靠视觉线索传达比较对象之间的关系[35].

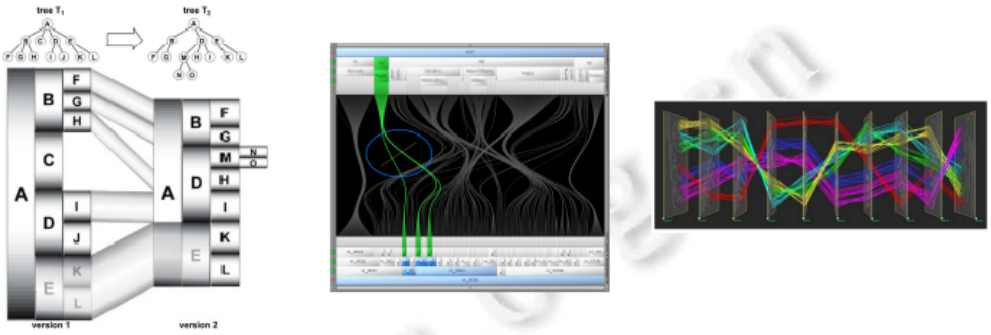
为用户提供视觉线索常用的方式有两种:一种是直接连接法,一种是视觉映射法.此外,矩阵也是一种特殊的并置比较方式.

© 中国科学院软件研究所 http://www.jos.org.cn

2.1.1 直接连接(direct linking)

直接连接法中,用连线将匹配节点连接,用以引导用户建立起视觉连接和数据实体连接的关系.而连线的透明度可以用来度量相似度,颜色可以编码类别等信息.

Chevalier 等人[18]使用带注解的语法树可视化 C++源代码的演变过程,以此来检测不同代码片段的变化.作者根据语义信息和结构信息将 C++代码整合成为树形结构,用连线连接代表相同代码片段的节点,通过比较来分析代码中哪些地方有改变,哪些地方没有改变.如图 2(a)所示,容易看出,代码段 C 和 J 在右侧版本中被删除而在右侧版本中新添加了代码段 M.



(a) 源代码比较[18] (b) 边捆绑技术[11] (c) 三维并置[34]

Fig.2 Juxtaposition-Direct linking

图 2 并置-直接连接

当采用直接连接法时,连线较多会使得整个可视化显示变得混乱,不易于数据的理解.边捆绑(hierarchical edge bundle,简称 HEB)技术是一种常用的处理方式,将位置上接近的连线进行一定程度的合并,能够减少连线过多造成的视觉混乱.周虹等人[48]详细介绍了边捆绑在信息可视化中的使用.Holten 等人[11]把边捆绑应用在源代码比较可视化中,使得两棵树结构的分离和偏移更加明显(如图 2(b)所示).Telead 等人[16]在他们的工作 CodeFlow 中,对 C++源代码进行可视化时,同样使用了 HEB 技术,用来比较不同代码中的结构差异.

Stewart 等人[34]将并置的比较方法应用在三维空间.他们使用了类似平行坐标的展开方式对生物学中的系谱树进行比较.如图 2(c)所示,每个切片代表一个系谱树,使用横向的连线将所有系谱树中的对应叶子节点进行连接,并用颜色表示这些节点所代表的类别.这种方法侧重于比较不同系谱树之间的关系,不足之处在于:由于采用了三维的可视化方法,用户无法方便地查看单个系谱树的结构特点,需要借助辅助视图分析单个系谱树.

2.1.2 视觉映射(visual mapping)

视觉映射法中,一般使用相同的颜色或者图像符号来表示两棵树中的对应节点,以此建立起两棵树之间的联系.

为了帮助生物学家理解不同系谱树之间的复杂关系,Munzner 等人[8]设计了 TreeJuxtaposer,将两棵系统发生树并列放置并计算节点的对应关系,根据计算结果高亮结构差异(如图 3 所示),同时辅以刷选(brushing)的交互方式,帮助用户寻找其中的相似部分.在 TreeJuxtaposer 中,用户可以动态查询符合某个条件的节点并高亮显示,以此来进行子结构的相似性比较.不同的系谱算法产生的系谱树具有相同的叶子节点,所以 Nye 等人[7]在比较不同的算法产生的系谱树时,依照叶子节点的对应关系对每个分支计算出一个拓扑分数,然后根据拓扑分数在分支中寻找最优的一对一匹配.将两棵树并列放置后,用相同的颜色编码对应分支.InfoZoom[36]将树结构数据转化成表格的表现形式,每个叶子节点被表示成一列,从根节点到叶子节点用行属性表示,将两棵树并行放置标示出不同的格子进行比较.在此基础上,通过缩放来解决当数据量较多时,每个单元格分配到的面积太小而无法识别的问题.而在 DoubleTree[10]中,采用上下结构放置两棵树,当用户选择其中一棵树的节点时,另一棵树中的对应节点也高亮显示,一个节点存在多个对应节点或不存在对应节点时,使用不同的高亮颜色.在图 4 中,当选择

© 中国科学院软件研究所 http://www.jos.org.cn

上方节点时,下方与之对应的节点会高亮显示.

Zhao 等人[35]在研究自然语言处理(NLP)中的语言解析器(discourse parser)时,为了改善算法的准确性,需要分析和比较不同的算法或者相同的算法不同参数的结果,为此提出了 DAViewer 的可视化原型.

DAViewer 中使用了基于修辞树(rhetorical structure tree)的文档组织结构可视化.根据算法的分词结果和词语的属性分类构建树形结构,将这些结果平铺放置来比较.DAViewer 使用了冰柱图(icicle plot)和系统树图(dendrogram)的结合体来表现树结构.将所有叶子节点对齐,扩张其他节点在冰柱图中的宽度,就可以很方便地进行列比较.在显示的树结构中,用节点的颜色编码相似度,冰柱图中节点的底色表示计算的错误率,这样,冰柱图同时可以作为一个热力图,表示算法在不同子结构上的计算精确度(如图 5 所示).

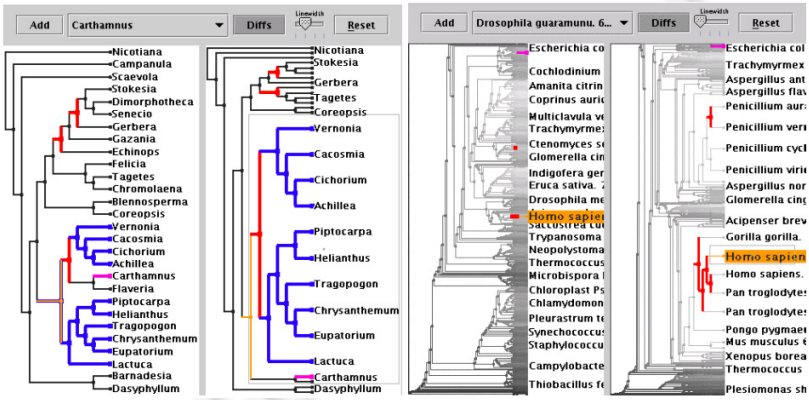


Fig.3 Using color to encode corresponding nodes, structure comparison mainly

图 3 用相同的颜色表示对应的节点,主要进行结构比较

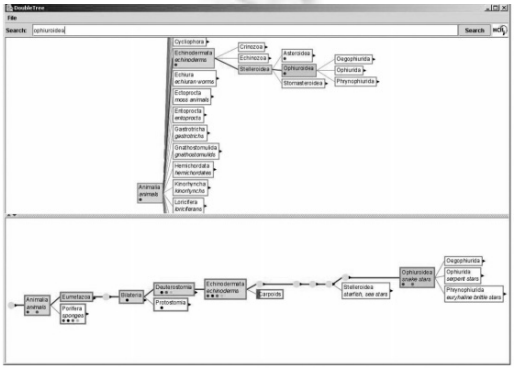


Fig.4 Double tree[10]

图 4 Double tree[10]

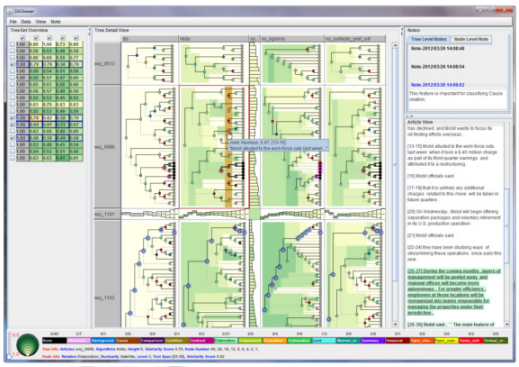


Fig.5 DAViewer[35]

图 5 DAViewer[35]

2.1.3 矩阵(matrix)

在树比较中,矩阵作为一种特殊的并置方法,有两种使用方式:一种是将节点平铺为矩阵的两个轴进行节点间的比较,另一种是作为总览图进行总体相似度的比较.前者多用于两树比较,而后者用于多树比较.

通过将比较的树节点作为矩阵的横轴和纵轴,在矩阵中可以比较对应节点的关系.如果将同一个树的节点作为横轴和纵轴,则可以进行树内比较.根据树的内部节点是否参与比较,可以选择缩进方法或者冰柱图[4].Ghoniem 等人[37]阐述了在进行大规模图可视化时矩阵方法的优势,矩阵方法侧重于对节点的比较,相对于其他方式,矩阵的方法欠缺对数据的结构比较[35].Melville 等人[38]分析了图可视化中基于矩阵的比较方法后得出:使用一个矩阵比较的方式能够提高可视化的效果以及寻找差异的准确性,优于用两个矩阵单独表示两个图的方

© 中国科学院软件研究所 http://www.jos.org.cn

式.同时,矩阵方式的局限性在于比较的树结构一般有相同的叶子节点集合.

Van Ham 等人[39]将矩阵方法用在软件系统不同模块之间的调用关系比较上,缺点是没有将节点之间的层次关系表现出来.Abello 等人[40]以及 Beck 等人[15]都使用了基于邻接矩阵的方式,将树结构作为矩阵的坐标轴,而将叶子节点的比较放在对应的矩阵元素中.不同的是,前者用于比较不同人种、国家发病率的层次分布,树结构采用点线链接的形式表示(如图 6 所示);后者用于可视化软件结构,树结构以冰柱图的方式给出,每个矩阵元素代表节点之间的依赖关系(如图 7 所示).

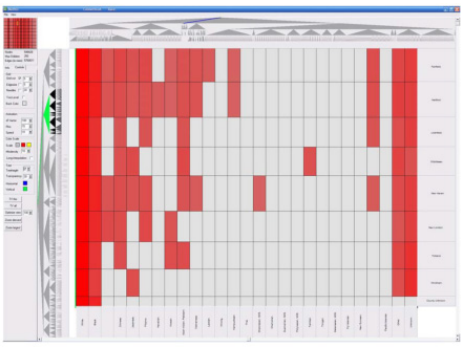


Fig.6 Matrix zoom[40]

图 6 矩阵缩放[40]

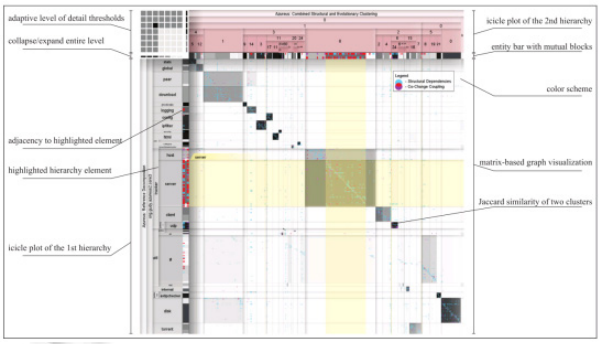


Fig.7 Comparison of software architecture[15]

图 7 软件结构比较[15]

Bremm 等人[41]使用矩阵视图作为总览图(图 8 左上方),列出了任意两棵树之间的相似度,进而帮助用户选择一部分树进行比较,然后通过并置的方式比较选出的树和基准之间的差异.

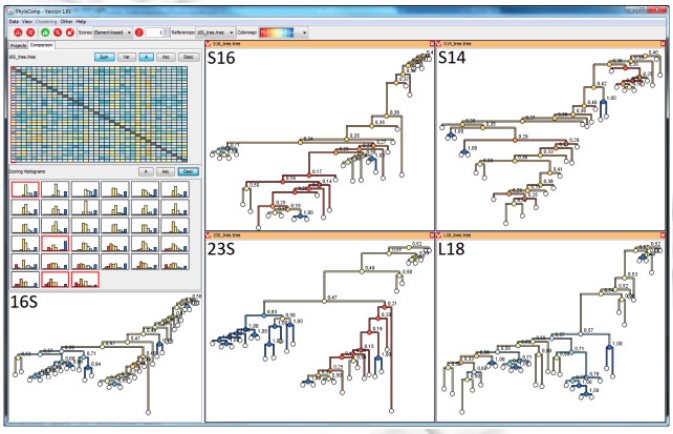


Fig.8 Matrix as overview[41]

图 8 矩阵作为总览图[41]

2.1.4 小 结

在并置的比较方法中,大多数采用点线链接形式的树可视化方法,这是因为树图会有以下的问题[13]:(1) 当树结构发生变化时,节点的位置可能发生突变,很难追踪节点的位置变化;(2) 缺少显著的可视模式呈现布局;(3) 缺少直接的对比方式来高亮差异;(4) 树图在结构的展示上有所欠缺.由于对应节点之间存在一定的距离,在对属性值的比较上稍显欠缺,所以并置的方法比较适合进行结构上的比较.同时,因为点线链接形式的树可视化方法本身具有空间利用率低的特点,在树的节点数量以及层数增多的时候,处在低层的节点数量呈指数级增长,这也就导致了并置的比较方法的本身扩展性不高.通过合适的交互,可以在一定程度上弥补这方面的不足.

© 中国科学院软件研究所 http://www.jos.org.cn

并置的方式多用于两树比较,通过平铺或向三维空间扩展的方式,可以用于多于两树的比较中.表 3 对并置的 3 种方法进行了比较.

Table 3 Summary of juxtaposition methods

表 3 并置方法总结

	优点	缺点	适用情况
直接连接	通过连线的密集程度判断节点对应关系以及相似度;注重总体的节点对应	连线较多时变得杂乱而影响视觉呈现	用于比较相似性,寻找其中的对应节点,而且节点数量不是很多的情况
视觉映射	通过视觉符号或颜色寻找节点对应,通常借助交互观察局部的对应关系	用户需要根据提供的视觉元素判断节点的对应关系	用视觉映射的方法发现其中的对应节点,进行结构上的比较
矩阵	可以进行一对多的节点关系比较	无法进行结构比较	多树比较时,可以用作总览图,为用户进一步选择提供依据

在并置的方式中,由于需要同时显示比较的树,常常导致空间不够或者节点所占空间太小而无法有效辨识.因此,折叠/展开以及缩放是并置中常用的交互方式[7,8,10].除此之外,在并置的比较方法中,尤其是采用直接连接的可视化系统[11,18,34],或者需要手动关联对应节点,或者在系统预先计算出对应节点连接后进行调整,都需要连接的交互方式.在采用视觉映射的方式时,使用颜色或者符号映射进行比较不如连线那么直观,这种情况下,通常会加入过滤/查询的交互[7,10],帮助用户快速地进行节点定位.

2.2 合 并

合并的方法是指采用某种算法计算比较树的节点之间的对应关系,根据对应关系将对应节点合并成一个树节点,进而构建一个包含所有节点的树结构.因此,合并后的树结构存在两类节点:合并节点和独立节点.在合并的节点内部,可以通过可视编码属性进行对应节点的比较,而独立节点代表独有的节点.因此,合并方法的核心在于通过直接比较被合并节点的属性值来看节点的属性差异,而通过“节点的有无”比较结构上的差异.

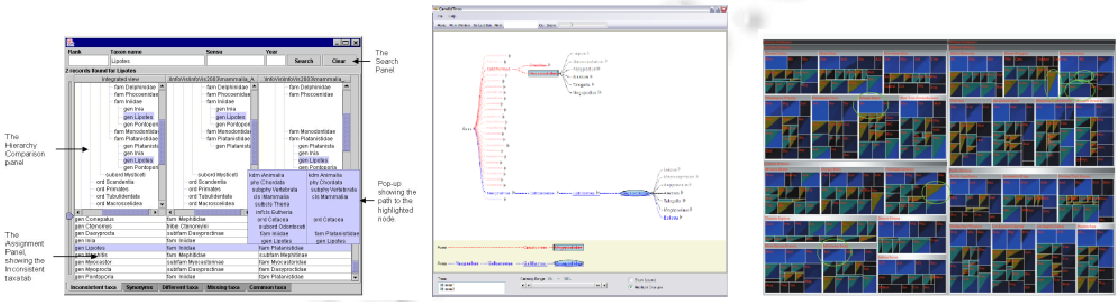
用并置的方式进行比较,用户需要不断地在比较对象之间切换视线,当关注的部分距离较远的时候,视线切换会非常频繁,增加理解的难度,可以通过合并或者重叠的方式来减少视线在比较对象上的切换.因此,合并的方法适用于比较的树结构相似度很高的时候,这样,通过合并对应的节点进行对应节点的属性值的比较.

根据在合并节点中的比较方式,合并的树比较方法分为直接比较和显式编码(explicit encoding)两种.

2.2.1 直接比较

直接比较是指在合并的节点内部直接可视编码对应的节点的属性的方式进行比较.

Morse 等人设计了 TaxoNote[9]对多类别层次结构划分进行比较.TaxoNote 将所有分类包含在同一个层次结构中,如图 9(a)所示,在展示合并后的结构时,采用了基于缩进的方式,对应的节点可以相邻放置在同一列中.TaxoNote 实现了简单的比较,将比较的两个对象放在合并后的对象两侧,同时,用一个表格列出所有类别信息,用户可以通过查询选择感兴趣的类别,从而在树结构中高亮,以此来达到比较的目的.



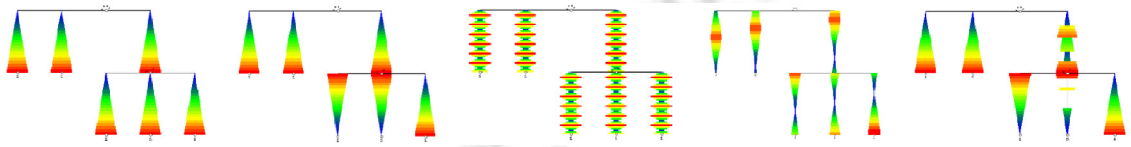
(a) TaxoNote[9] (b) CandidTree[43] (c) 对比树图[13]

Fig.9 Compare directly

图 9 直接比较

Zoomology[42]用于比较两个分类数据集,由于数据的结构相似度很高,采用的是将两棵树合并成一个冰柱概览图的方法,主要比较数据的结构差异.Lee 等人[43]将层次结构的结构差异称为不确定性,实现了 CandidTree (如图 9(b)所示)来比较两个相似的层次结构中的不确定性.CandidTree 将两个树合并成一个,分别使用颜色和透明度编码差异.例如,在两棵树中都出现的节点、只存在于第 1 棵树中的节点以及只存在于第 2 棵树中的节点分别用 3 种不同的颜色表示.用户根据颜色来判断相似的部分以及差异较大的部分,同时,可以交互地比较不同深度、不同层次的节点.当用户选择一个节点时,显示从根节点到当前节点的路径,进而比较绝对路径的差异.他们将 CandidTree 应用在目录结构的比较上,可以很方便地看到目录结构的差异,比如文件的添加、删除以及移动,如图 9(c)所示.Goemez 等人[12]也使用了相同的颜色编码的方式.

Burch 等人[49]使用正交的直线连接父子节点,在延伸的直线上展示时序数据,通过颜色和厚度编码属性信息,在一个节点中同时显示不同时刻的属性.图 10 显示了节点属性的 5 种变化趋势.Ghoniem 等人[50]设计了动画树图,将动画的效果应用在树图中,用以观察树节点的属性变化差异.因为树图的节点所占面积和权重对应,所以在动画过程中,用户可以集中在关注区域的变化上,而且面积的变化可以平滑过渡.



(a) 正常趋势 (b) 逆趋势 (c) 周期变化 (d) 间歇性变化 (e) 不规则变化

Fig.10 Five types of value behavior[49]

图 10 5 种不同的属性值变化[49]

Tu 等人[13]在进行树比较时,同样使用了合并的策略.不同的是,他们采用了树图作为树的表现形式.树图的优势在于,它能够在反映数据属性总体变化的同时呈现个体的属性的变化,这是树图的结构所决定的.在树图中,父节点的属性值是其子节点属性的总和,这个特点在使用树图可视化文件目录结构的时候尤其明显.他们将两个树图合并,具有相同父节点的元素被合并到同一个树图单元中,从而可以直接比较两个实体的属性.为了降低布局突变的影响,他们为此还提出了螺旋树图(spiral treemap)布局算法.如图 9(c)所示,对 NBA 中 2002~2003 和 2003~2004 两个赛季的球员所在球队的统计进行了可视化,使用节点的两个对角区域编码两个赛季中球员的球队归属,用所占比例表示两个赛季的得分情况.使用树图的形式,虽然可以使用颜色和纹理表示节点值的差异,但是无法表现内部节点结构上的差异.

其他一些工作[14,19,20,44,47]在进行树比较时,也都使用了合并的方法.Ambros 等人[19]将合并比较用于代码版本管理系统中 Bug 的追踪分析,Timothy[20]用于分析文档结构随时间的变化,比较不同时间点的文档结构差异,从而为数字鉴证人员提供了一个分析工具.

2.2.2 显式编码(explicit encoding)

在合并的比较方法中,另一种方式是显式编码.显式编码是指直接可视化节点属性的关系和值的不同,比如差值、变化百分比例等.Gleicher 等人[3]把显式编码作为一种独立的比较方法,但是在树结构中,比较不能完全脱离结构,所以通常用在合并以后的树结构中.显式编码侧重于可视化数据在结构或者属性上值的差异,弱化了对结构和属性绝对值的可视化比较.显式编码的好处是,在数据量增大的时候,用户可以很容易地注意到数据中变化比较大的节点[3].

TreeVersity2[14]用于进行层次数据的时变比较,一般是层次化后的表格数据.比如肺癌死亡率的数据,按照种族、性别以及是否吸烟分组后的结果.在时间线视图中,分层显示节点属性随时间的变化,Stem 视图用于比较两个时间点的数据,又分为两个子视图:点线子视图用点线连接的形式,每个节点用颜色、高度、形状等表示不同属性的差异值;柱状子视图将柱状图和冰柱图结合,层之间用邻接关系表示父子关系,每个节点用颜色、高度、宽度、方向等编码不同属性的差值或变化百分比(如图 11 所示).在比较任务中,颜色编码是经常用到的一种方

© 中国科学院软件研究所 http://www.jos.org.cn

式,在一些情况下,多个属性需要全局的颜色映射,但是在属性的值域差异很大时,值域范围小的属性几乎不可见.Tominski 等人[51]提出用合并一部分重叠值的方式来改善比较任务中的颜色编码.

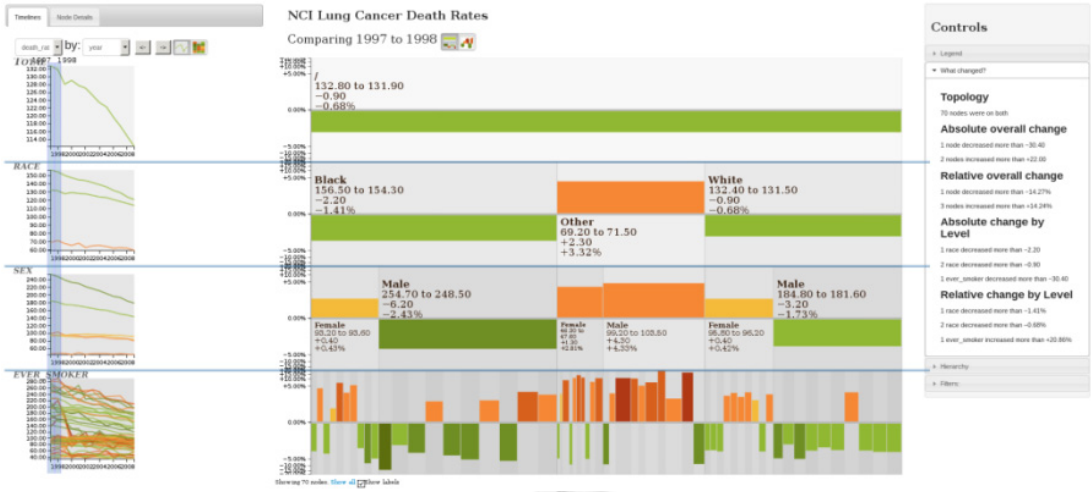


Fig.11 Explicit encoding, TreeVersity2[14]

图 11 显式编码,TreeVersity2[14]

2.2.3 小 结

合并的比较方法是利用了层次数据在结构上的相似性而采取的数据比较策略,多用于比较的树相似性很高时.直接比较用于比较属性的绝对值差异,而显式编码用于比较相对值差异,结构性的差异可以通过各自独有的节点数目来得出.合并的方式一般用于两树比较中.表 4 总结了两种合并的比较方法的优缺点以及适用情况.

采用合并的树比较方法,比较的信息大多集中在合并节点的内部,因此,交互多以能够有助于查看细节的方式为主,比如简单的缩放(zooming)、焦点+上下文(focus+context)技术[43]、鱼眼(fish eye)技术[52]等.

Table 4 Summary of superposition methods

表 4 合并方法总结

名称	优点	缺点	适用情况
直接比较	注重属性值的绝对量之间的比较	需要同时可视化比较节点的属性,空间使用较多	进行对应节点属性值的直接比较
显式编码	属性的相对值比较,注重比较值的差异程度	缺少属性的绝对量的上下文,可能会造成数据的错误理解	当属性值的绝对量不是关注的重点时,可以考虑使用显式编码的方式

2.3 动 画

动画的方式常用于动态树的比较中,尤其是具有实变特征的数据,其目的在于发现数据在结构或属性值上的变化趋势.动画能够有效地表现结构上的变化[53]和属性的变化趋势[54].Card 等人[45]设计了 TimeTree(如图 12 所示),将树可视化应用在时变数据中,不同时间点的树结构变化使用了动画来展现,通过拖动底部的一个时间轴的方式实现不同的时间点视图的切换.Tu 等人[13]采用将树图和地图联动的方式,通过地图的纹理变化查看数据的变化程度,如图 13 所示.

动画的方式除了用在动态树的比较之外,还可以用在两树以及多树的比较上,所采取的方式是在比较的树之间做动画切换.Robertson 等人[54]在进行 Polyarch 可视化时加入了动画的效果,用以观察不同树结构中节点的增加和减少.不稳定的树图布局算法,导致动画的效果比较差[55].Tu 等人[13]在进行树比较时提出了树图的螺旋式布局(spiral layout).螺旋式的布局方式在很大程度上提高了布局的连续性和稳定性.同样是基于布局的稳定性考虑, Gotz[46]提出了 Dynamic Voronoi Treemap 算法,能够保持稳定性和良好的纵横比,能够平滑、实时地展示

© 中国科学院软件研究所 http://www.jos.org.cn

时序变化的层次数据.这些方法在一定程度上降低了不稳定的布局算法对动画效果的影响,在文献[52,56,57]中也都使用了动画的方式进行树的比较.

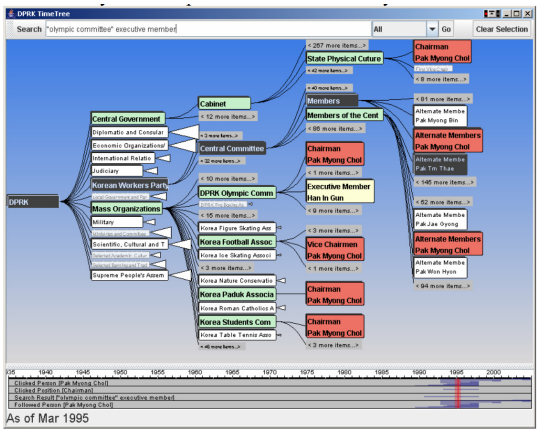


Fig.12 TimeTree[45]

图 12 TimeTree[45]

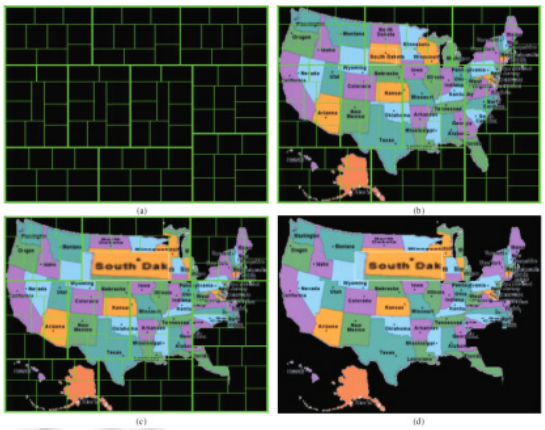


Fig.13 Using image texture changes to reflect changes in data[13]

图 13 用地图的纹理变化反映数据变化[13]

由于在动画过程中树中节点的位置会发生变化,用户不能很好地跟踪树中变化的部分[4].尽管在平滑过度的动画中用户可以跟踪到变化的位置,但是一个时刻用户只能看到比较的树中的一棵并关注局部位置,这就增加了比较的难度.所以,动画的方式相对其他比较方式来说使用较少.

近年来,有不少工作将其他方法引入到树的比较可视化中.Amenta 等人[6]以及 Hillis 等人[57]都采用了 MDS (multi-dimensional scaling)技术,根据多维属性计算任意两棵树之间的距离,然后将其映射到一个散点图中.通过这种方式,实现了多树的比较.这类方法只能在粗粒度层次上对树进行分类比较,而无法进行细节性的比较.

3 树比较可视化中的交互

可视化是帮助用户建立数据的构思模型(mental models)的工具[58−61],而交互允许用户与数据快速交流,更好地验证假设和发现内在联系,帮助用户调整构思模型[62].交互注重探索的过程,在某种程度上能够弥补数据展示方式的不足.任磊等人[63]将人机交互技术概括为 5 类:动态过滤技术(dynamic query)与动态过滤用户界面、整体+详细技术(overview+detail)与 Overview+Detail 用户界面、平移+缩放技术(panning+zooming)与可缩放用户界面(ZUI)、焦点+上下文技术与 Focus+Context 用户界面、多视图关联协调技术(multiple coordinated view)与关联多视图用户界面.

在可视比较中,交互一般有两种目的:(1) 连接相关的对象(连接,高亮);(2) 重新排列和整理对象[47].在树可视化中,折叠展开、缩放以及查询是常用的交互方式,这些方式也常用在树比较可视化中.通过折叠展开和缩放,能够在节省空间的同时让用户关注于重要的部分,而查询能够进行节点的快速定位.在树比较的可视化方法中,连接(linking)和刷选(brushing)的交互方式常用于标记数据中用户感兴趣的部分.Becker 等人[64]指出,在用并置的方式做数据比较时,常常结合连接和刷选操作,尤其是在多个视图协作的情况下.Graham 等人[65]提出,将连接操作和增强焦点概念应用到树形结构中.在树结构中,用户可以交互地选择相似的节点进行连接或高亮操作,而这些节点较之其他节点更受用户的关注,因此,这些节点会分配到更多的空间,而用户关注的焦点就会更多地集中在这些节点上.采用这种方式,可以突出相似的节点.如图 14 所示,Kehrere 等人[66]在进行树内比较时,通过用线连接比较的对象来指出比较的方向.为了比较一棵树中不同的子树,Burch[67]在树图中引入了相似链接的概念,用户可以根据节点属性以及所在的层次,动态地将相似节点连接起来.同时,用户可以自定义不同属性的阈

© 中国科学院软件研究所 http://www.jos.org.cn

值以及层次表示.

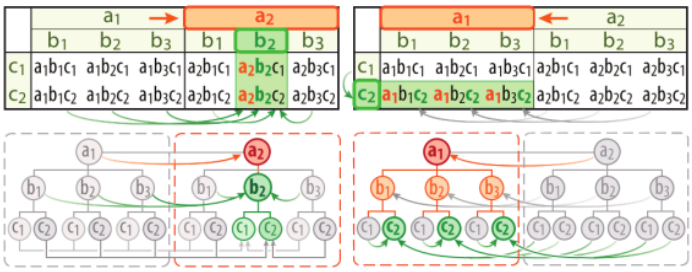


Fig.14 Compare direction by linking[66]

图 14 用连线指明比较的方向[66]

Munzner 等人[8]引入了可见性保证(guaranteed visibility)的概念,其核心思想是:在缩放和折叠操作中,时刻保持用户关注部分的可见性,这就使得关注的相似节点在突出的位置显示.Conklin 等人[68]将分解可视化的方式(breakdown visualization)应用在表格数据和统计数据中,允许用户按照不同的层次将数据分割成不同的分组,同时允许用户在一棵树中的不同分支内比较.TreeJuxtaposer[8]中使用了 Focus+Context 的技术,在保留概览的情况下进行局部的比较.任磊等人[69]综述了海量层次信息可视化与 Focus+Context 技术的相关工作,针对海量层次信息可视化的交互问题,在嵌套圆可视化技术的基础上提出了基于上下文感知的 Focus+Context 交互式可视化技术.InfoZoom[36]中则使用缩放的手段显示低层的节点.而在 Chevalier 等人[18]的工作中,引入了细节层次(level of detail)的概念,用户可以交互地选择显示哪些层次的细节.Isenberg 等人[70]将树比较的方法应用在多人协作中,将多点触控的屏幕作为显示媒介,结合多点触控屏幕的交互优势,用户能够更加方便地进行比较.Tominski 等人[47]受到现实生活中人们进行的各种比较操作的启发,提出可以通过附加视觉线索来支持可视比较.在实现的交互比较的原型中,通过折叠的交互方式来解决节点重合带来的覆盖问题,用户可以根据需要将上层节点折叠,使得下方的节点能够显示出来.Plumlee 等人[71]指出,在进行比较任务时,通过添加注解的方式有助于降低用户的理解成本.表 5 总结了树比较可视化中常用的交互方式以及其作用.

Table 5 Interactive techniques in visualization of tree comparison

表 5 树比较可视化中的交互

类别	交互方式	目的/作用
基本交互	缩放	进行局部位置节点的细节查看
	折叠/展开	显示或者隐藏部分层次节点
	连接	建立节点之间的对应关系
	刷选	选择用户感兴趣的部分进行进一步比较操作
	查询/过滤	快速定位节点
焦点相关的交互	可见性保证	在视图改变过程中将感兴趣的节点一直保留在视图中
	总览+细节	在一个全局图中进行细节的比较查看
	焦点+上下文	在进行细节比较的同时为用户保留当前位置在全局中的上下文信息

4 树比较可视化面临的挑战以及发展

树比较的可视化方法主要面临以下两个方面的挑战.

(1) 树结构的表现形式以及比较形式

树比较的表现形式受到树的表现形式的约束,常用的树表现形式为缩进形式、点线链接形式、邻接形式和树图.从比较形式上说,树比较可视化方法以并置和合并两种方式为主,而这也是符合人比较习惯的方式.无论是发掘或创造新的树表现形式还是比较形式,在可视化领域中都是一个比较难的问题.

(2) 扩展性

© 中国科学院软件研究所 http://www.jos.org.cn

树的结构决定了当树节点以及层次增多的时候会带来扩展性问题,同时,对比较的形式也会产生影响.随着数据量越来越大、比较任务的不断变化以及不同数据的比较需求,现有方法的局限性越来越明显.一方面需要根据可视化的需求选择合适的方法,另一方面也需要在现有的方法基础上做一些创新性的改变.

目前的树比较可视化方法中,大多采用的是点线链接形式和树图形式,其他形式相对较少.而用到的点线链接形式中,又以从左到右/从上到下的布局为主,空间利用率不高,其他布局方式用到的很少.从 Schulz 等人[32]提出的树可视化方法设计空间出发,结合比较任务的特点,可以进行一些方法上的尝试,比如使用径向布局的方式.而陈谊等人[72]在分析关联层次数据时,提出了 SR(sunburst with relationship)算法,通过将层次节点聚集的方法应用于 Sunburst 中,为可视化关联关系提供了空间.该算法可以用于树比较中,在保持层次结构上下文的情况下对节点的属性进行可视比较.随着科技的发展,除了电脑屏幕以外,可视化也开始应用在诸如手机、平板以及大屏等设备上,这些设备具有不同的显示特性以及操作特性.如何让现有的可视化技术适应新的设备,以及在这些设备上如何进行交互,都是值得研究的课题.

5 总 结

本文对树比较可视化中使用的方法进行了总结,树比较可视化方法按照表现形式分为并置、合并以及动画这 3 种.并置的方法多用于进行树的结构比较;合并的方法适合比较节点的属性;动画的方式则用于具有时序性质的层次数据,比较数据不同时间点的差异以及数据随时间的变化趋势.综合以上分析,表 6 列出了不同方法的比较,作为树比较可视化方法选择的参考.同时,本文还对树比较可视化中的交互以及该研究领域面临的挑战和发展进行了讨论.

Table 6 Summary of tree comparison visualization

表 6 树比较可视化方法总结

		树内比较	两树比较	多树比较	静态树	动态树	属性比较	结构比较
并置	直接连接	–	★★★★★	★☆☆☆☆	★★★★☆	★☆☆☆☆	★★★☆☆	★★★★☆
	视觉映射	–	★★★★★	★★☆☆☆	★★★★☆	★☆☆☆☆	★★★★☆	★★★☆☆
	矩阵	★★☆☆☆	★★☆☆☆	★★★☆☆	★★★★☆	★☆☆☆☆	★★★☆☆	★☆☆☆☆
合并	直接比较	–	★★★★★	★★☆☆☆	★★★★☆	★☆☆☆☆	★★★★☆	★★★☆☆
	显式编码	–	★★★★★	★☆☆☆☆	★★★★☆	★☆☆☆☆	★★★★☆	★★★☆☆
动画		★★☆☆☆	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★★★☆	★★★☆☆	★★★☆☆

注:本表可以作为树比较可视化方法选择的基本依据,在实际选择中,需要结合数据特点以及可视化目的进行选择

致谢 本文在撰写过程中得到了北京大学智能科学系袁晓如研究员的指导,并与其学生进行了多次讨论,在此表示感谢.

References:

[1] Cheng XQ, Jin XL, Wwang YZ, Guo JF, Zhang TY, Li GJ. Survey on big data system and analytic techenology. Ruan Jian Xue Bao/Journal of Software, 2014,25(9):1889−1908 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4674.html [doi: 10.13328/j.cnki.jos.004674]

[2] Card SK, Mackinlay JD, Shneiderman B. Readings in Information Visualization: Using Vision to Think. San Francisco: Morgan Kaufmann Publishers, 1999. 1−712.

[3] Gleicher M, Albers D, Walker R, Jusufi I, Hansen C, Roberts J. Visual comparison for information visualization. Information Visualization, 2011,10(4):289−309. [doi: 10.1177/1473871611416549]

[4] Graham M, Kennedy J. A survey of multiple tree visualization. Information Visualization, 2010,9(4):235−252. [doi: 10.1057/ivs.2009.29]

[5] Marco A, Marín I. A general strategy to determine the congruence between a hierarchical and a non-hierarchical classification. BMC Bioinformatics, 2007,8(1):442. [doi: 10.1186/1471-2105-8-442]

© 中国科学院软件研究所 http://www.jos.org.cn

[6] Amenta N, Klingner J. Case study: Visualizing sets of evolutionary trees. In: Chuang WP, ed. Proc. of the IEEE Symp. on Information Visualization. Boston: IEEE Computer Society Press, 2002. 71−74. [doi: 10.1109/INFVIS.2002.1173150]

[7] Nye TM, Liò P, Gilks WR. A novel algorithm and Web-based tool for comparing two alternative phylogenetic trees. Bioinformatics, 2006,22(1):117−119. [doi: 10.1093/bioinformatics/bti720]

[8] Munzner T, Guimbretière F, Tasiran S, Zhang L, Zhou YH. Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. ACM Trans. on Graphics, 2003,22(3):453−462. [doi: 10.1145/882262.882291]

[9] Morse DR, Ytow N, Roberts DM, Sato A. Comparison of multiple taxonomic hierarchies using TaxoNote. In: Plaisant C, ed. Proc. of the Compendium of Symp. on Information Visualization. Washington: IEEE Computer Society Press, 2003. 126−127.

[10] Parr CS, Lee B, Campbell D, Bederson BB. Visualizations for taxonomic and phylogenetic trees. Bioinformatics, 2004,20(17): 2997−3004. [doi: 10.1093/bioinformatics/bth345]

[11] Holten D, van Wijk JJ. Visual comparison of hierarchically organized data. Computer Graphics Forum, 2008,27(3):759−766. [doi: 10.1111/j.1467-8659.2008.01205.x]

[12] Guerra-Gomez JA, Buck-Coleman A, Plaisant C, Shneiderman B. TreeVersity: Comparing tree structures by topology and node's attributes differences. In: Miksch S, ed. Proc. of the IEEE Conf. on Visual Analytics Science and Technology. Providence: IEEE Computer Society Press, 2011. 275−276. [doi: 10.1109/VAST.2011.6102471]

[13] Tu Y, Shen HW. Visualizing changes of hierarchical data using treemaps. IEEE Trans. on Visualization and Computer Graphics, 2007,13(6):1286−1293. [doi: 10.1109/TVCG.2007.70529]

[14] Guerra-Gomez JA, Pack ML. Visualizing change over time using dynamic hierarchies: TreeVersity2 and the StemView. IEEE Trans. on Visualization and Computer Graphics, 2013,19(12):2566−2575. [doi: 10.1109/TVCG.2013.231]

[15] Beck F, Diehl S. Visual comparison of software architectures. Information Visualization, 2013,12(2):178−199. [doi: 10.1177/1473871612455983]

[16] Telea A, Auber D. Code flows: Visualizing structural evolution of source code. Computer Graphics Forum, 2008,27(3):831−838. [doi: 10.1111/j.1467-8659.2008.01214.x]

[17] Wettel R, Lanza M. Visual exploration of large-scale system evolution. In: Zaidman A, ed. Proc. of the 15th Working Conf. on Reverse Engineering. Antwerp: IEEE Computer Society Press, 2008. 219−228. [doi: 10.1109/WCRE.2008.55]

[18] Chevalier F, Auber D, Telea A. Structural analysis and visualization of C++ code evolution using syntax trees. In: Proc. of the 9th Int'l Workshop on Principles of Software Evolution: Conjunction with the 6th ESEC/FSE Joint Meeting. Dubrovnik: ACM Press, 2007. 90−97. [doi: 10.1145/1294948.1294971]

[19] Ambros MD, Lanza M. Software bugs and evolution: A visual approach to uncover their relationship. In: Proc. of the 10th European Conf. on Software Maintenance and Reengineering. Bari: IEEE Computer Society Press, 2006. 229−238. [doi: 10.1109/CSMR.2006.51]

[20] Leschke TR, Nicholas C. Change-link 2.0: A digital forensic tool for visualizing changes to shadow volume data. In: Goodall J, ed. Proc. of the 10th Workshop on Visualization for Cyber Security. Atlanta: ACM Press, 2013. 17−24. [doi: 10.1145/2517957.2517960]

[21] Boardman R. Bubble trees the visualization of hierarchical information structures. In: Proc. of the Extended Abstracts on Human Factors in Computing Systems (CHI 2000). Hague: ACM Press, 2000. 315−316. [doi: 10.1145/633292.633484]

[22] Grivet S, Auber D, Domenger JP, Melancon G. Bubble tree drawing algorithm. In: Wojciechowski K, ed. Proc. of the Computer Vision and Graphics. Warsaw: Springer-Verlag, 2006. 633−641. [doi: 10.1007/1-4020-4179-9_91]

[23] Johnson B, Shneiderman B. Tree-Maps: A space-filling approach to the visualization of hierarchical information structures. In: Nilson GM, ed. Proc. of the 2nd Int'l IEEE Visualization Conf. San Diego: IEEE Computer Society Press, 1991. 284−291. [doi: 10.1109/VISUAL.1991.175815]

[24] van Wijk JJ, de Wetering HV. Cushion treemaps: Visualization of hierarchical information. In: Wills G, ed. Proc. of the IEEE Symp. on Information Visualization. San Francisco: IEEE Computer Society Press, 1999. 73−78. [doi: 10.1109/INFVIS.1999.801860]

[25] Bruls M, Huizing K, van Wijk JJ. Squarified treemaps. In: de Leeuw W, ed. Proc. of the Joint EUROGRAPHICS and IEEE TCVG Symp. on Visualization (Data Visualization 2000). Amsterdam: Springer-verlag, 2000. 33−42. [doi: 10.1007/978-3-7091-6783-0_4]

© 中国科学院软件研究所 http://www.jos.org.cn

[26] Shneiderman B, Wattenberg M. Ordered treemap layouts. In: Proc. of the IEEE Symp. on Information Visualization. IEEE Computer Society Press, 2001. 73−78. [doi: 10.1109/INFVIS.2001.963283]

[27] Jürgensmann S, Schulz HJ. Poster: A visual survey of tree visualization. In: Proc. of the IEEE Information Visualization. Salt Lake City: IEEE Society Press, 2010. 7.

[28] Xiao WD, Sun Y, Zhao X, Zhou C, Feng XS. A survey on hierarchical data visualization techniques. Journal of Chinese Computer Systems, 2011,32(1):137−146 (in Chinese with English abstract).

[29] Schulz HJ. Treevis.net: A tree visualization reference. IEEE Computer Graphics and Applications, 2011,31(6):11−15. [doi: 10.1109/MCG.2011.103]

[30] Shneiderman B. Treemaps for space-constrained visualization of hierarchies. 2014. http://www.cs.umd.edu/hcil/treemap-history/

[31] Zhang X, Yuan XR. Treemap visualization. Journal of Computer-Aided Design & Computer Graphics, 2012,24(9):1113−1124 (in Chinese with English abstract).

[32] Schulz HJ, Hadlak S, Schumann H. The design space of implicit hierarchy visualization: A survey. IEEE Trans. on Visualization and Computer Graphics, 2011,17(4):393−411. [doi: 10.1109/TVCG.2010.79]

[33] Pagendarm HG, Post FH. Comparative visualization: Approaches and examples. Reports of the Faculty of Technical Mathematics and Informatics. Delft University of Technology, 1995.

[34] Stewart CA, Hart D, Berry DK, Olsen GJ, Wernert EA, Fischer W. Parallel implementation and performance of FastDNAml: A program for maximum likelihood phylogenetic inference. In: Proc. of the ACM/IEEE Conf. on Supercomputing. 2001. 22−32. [doi: 10.1145/582034.582054]

[35] Zhao J, Chevalier F, Collins C, Balakrishnan R. Facilitating discourse analysis with interactive visualization. IEEE Trans. on Visualization and Computer Graphics, 2012,18(12):2639−2648. [doi: 10.1109/TVCG.2012.226]

[36] Spenke M, Beilken C. Visualization of trees as highly compressed tables with InfoZoom. In: Munzner T, ed. Proc. of the IEEE Symp. on Information Visualization. Seattle: IEEE Computer Society Press, 2003. 122−123.

[37] Ghoniem M, Fekete J, Castagliola P. A comparison of the readability of graphs using node-link and matrix-based representations. In: Wart M, ed. Proc. of the IEEE Symp. on Information Visualization. Austin: IEEE Computer Society Press, 2004. 17−24. [doi: 10.1109/INFVIS.2004.1]

[38] Melville AG, Graham M, Kennedy JB. Combined vs. separate views in matrix-based graph analysis and comparison. In: Banissi E, ed. Proc. of the IEEE Int'l Conf. on Information Visualization. London: IEEE Computer Society Press, 2011. 53−58. [doi: 10.1109/IV.2011.49]

[39] van Ham F. Using multilevel call matrices in large software projects. In: Munzner T, ed. Proc. of the IEEE Symp. on Information Visualization. Seattle: IEEE Computer Society Press, 2003. 227−232. [doi: 10.1109/INFVIS.2003.1249030]

[40] Abello J, van Ham F. Matrix Zoom: A visual interface to semi-external graphs. In: Wart M, ed. Proc. of the IEEE Symp. on Information Visualization. Austin: IEEE Computer Society Press, 2004. 183−190. [doi: 10.1109/INFVIS.2004.46]

[41] Bremm S, von Landesberger T, Hess M, Schreck T, Weil P, Hamacherk K. Interactive visual comparison of multiple trees. In: Miksch S, ed. Proc. of the IEEE Conf. on Visual Analytics Science and Technology. Providence: IEEE Computer Society Press, 2011. 31−40. [doi: 10.1109/VAST.2011.6102439]

[42] Hong JY, D'Andries J, Richman M, Westfall M. Zoomology: Comparing two large hierarchical trees. In: Keahey A, ed. Proc. of the Posters Compendium of Information Visualization. Seattle: IEEE Society Press, 2003. 120−121.

[43] Lee B, Robertson GG, Czerwinski M. CandidTree: Visualizing structural uncertainty in similar hierarchies. Information Visualization, 2007,6(10):233−247.

[44] Leschke TR, Sherman AT. Change-Link: A digital forensic tool for visualizing changes to directory trees. In: Proc. of the 9th Int'l Symp. on Visualization for Cyber Security. Seattle: ACM Press, 2012. 48−55. [doi: 10.1145/2379690.2379697]

[45] Card SK, Suh B, Pendleton BA, Heer J, Bodnar JW. Time tree: Exploring time changing hierarchies. In: Chung WP, ed. Proc. of the IEEE Symp. on Visual Analytics Science and Technology. Baltimore: Computer Society Press, 2006. 3−10. [doi: 10.1109/VAST.2006.261450]

[46] Gotz D. Dynamic voronoi treemaps: A visualization technique for time-varying hierarchical data. Technical Report, RC25132, IBM, 2011.

© 中国科学院软件研究所 http://www.jos.org.cn

[47] Tominski C, Forsell C, Johansson J. Interaction support for visual comparison inspired by natural behavior. IEEE Trans. on Visualization and Computer Graphics, 2012,18(12):2719−2728. [doi: 10.1109/TVCG.2012.237]

[48] Zhou H; Xu PP; Yuan XR, Qu HM. Edge bundling in information visualization. Tsinghua Science and Technology, 2013,18(2): 145−156. [doi: 10.1109/TST.2013.6509098]

[49] Burch M, Weiskopf D. Visualizing dynamic quantitative data in hierarchies. In: Csurka G, ed. Proc. of the Int'l Conf. on Information Visualization Theory and Applications. Vilamoura: SciTe Press, 2011. 177−186.

[50] Ghoniem M, Fekete J. Animating treemaps. In: Proc. of the 18th HCIL Symp. Workshop on Treemap Implementations and Applications. 2001.

[51] Tominski C, Fuchs G, Schumann H. Task-Driven color coding. In: Banissi E, ed. Proc. of the IEEE Int'l Conf. on Information Visualization. London: IEEE Computer Society Press, 2008. 373−380. [doi: 10.1109/IV.2008.24]

[52] Card SK, Nation D. Degree-of-Interest trees: A component of an attention-reactive user interface. In: Marsico MD, ed. Proc. of the Working Conf. on Advanced Visual Interfaces. Gallipoli: ACM Press, 2002. 231−245. [doi: 10.1145/1556262.1556300]

[53] Graham M, Kennedy JB, Hand C. A comparison of set-based and graph-based visualizations of overlapping classification hierarchies. In: Proc. of the Working Conf. on Advanced Visual Interfaces. Palermo: ACM Press, 2000. 41−50. [doi: 10.1145/345513.345243]

[54] Robertson G, Cameron K, Czerwinski M, Robbins D. Animated visualization of multiple intersecting hierarchies. Information Visualization, 2002,1(1):50−65. [doi: 10.1057/palgrave.ivs.9500002]

[55] Von Landesberger T, Kuijper A, Schreck T, Kohlhammer J, van Wijk JJ, Fekete JD, Fellner DW. Visual analysis of large graphs: State-of-the-Art and future research challenges. Computer Graphics Forum, 2011,30(6):1719−1749. [doi: 10.1111/j.1467-8659.2011.01898.x]

[56] Heer J, Card SK. DOITrees revisited: Scalable, space-constrained visualization of hierarchical data. In: Proc. of the Working Conf. on Advanced Visual Interfaces. Gallipoli: ACM Press, 2004. 421−424. [doi: 10.1145/989863.989941]

[57] Hillis DM, Heath TA, John KS. Analysis and visualization of tree space. Systematic Biology, 2005,54(3):471−482. [doi: 10.1080/10635150590946961]

[58] Fekete J. Advanced interaction for information visualization. In: North S, ed. Proc. of the IEEE Pacific Visualization Symp. IEEE Society Press, 2010. 9−13. [doi: 10.1109/PACIFICVIS.2010.5429617]

[59] Pike WA, Stasko JT, Chang R, O'Connell TA. The science of interaction. Information Visualization, 2009,8(4):263−274. [doi: 10.1057/ivs.2009.22]

[60] Cook KA, Thomas JJ. Illuminating the path: The research and development agenda for visual analytics. Technical Report, Richland: Pacific Northwest National Laboratory, 2005.

[61] van Wijk JJ. The Value of Visualization. IEEE Visualization, 2005. 79−86. [doi: 10.1109/VISUAL.2005.1532781]

[62] Yuan XR, Zhang X, Xiao H, Guo HQ. Visualization research frontier and prospect. e-Science Technology & Application, 2011,2(4): 3−13 (in Chinese with English abstract).

[63] Ren L, Du Y, Ma S, Zhang XL, Dai GZ. Visual analytics towards big data. Ruan Jian Xue Bao/Journal of Software, 2014,25(9): 1909−1936 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4645.html [doi: 10.13328/j.cnki.jos.004645]

[64] Becker RA, Cleveland WS. Brushing scatterplots. Technometrics, 1987,29(2):127−142. [doi: 10.1080/00401706.1987.10488204]

[65] Graham M, Kennedy J. Combining linking and focusing techniques for multiple hierarchy visualization. In: Elanissi E, ed. Proc. of the 5th Int'l Conf. on Information Visualization. Los Alamitos: IEEE Computer Society, 2001. 425−432. [doi: 10.1109/IV.2001.942092]

[66] Kehrer J, Piringer H, Berger W, Groller ME. A model for structure-based comparison of many categories in small-multiple displays. IEEE Trans. on Visualization and Computer Graphics, 2013,19(12):2287−2296. [doi: 10.1109/TVCG.2013.122]

[67] Burch M. Interactive similarity links in treemap visualizations. In: Banissi E, ed. Proc. of the IEEE Int'l Conf. on Information Visualization. Paris: IEEE Computer Society Press, 2014. 34−39. [doi: 10.1109/IV.2014.35]

[68] Conklin N, Prabhakar S, North C. Multiple foci drill-down through tuple and attribute aggregation polyarchies in tabular data. In: Chung WP, ed. Proc. of the IEEE Symp. on Information Visualization. Boston: IEEE Computer Society, 2002. 131−134. [doi: 10.1109/INFVIS.2002.1173158]

© 中国科学院软件研究所 http://www.jos.org.cn

[69] Ren L, Wang WX, Teng DX, Ma CX, Dai GZ, Wang HA. Focus+Context technique for interactive visualization of large hierarchies. Ruan Jian Xue Bao/Journal of Software, 2008,19(11):3073−3082 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/3073.htm

[70] Isenberg P, Carpendale S. Interactive tree comparison for co-located collaborative information visualization. IEEE Trans. on Visualization and Computer Graphics, 2007,13(6):1232−1239. [doi: 10.1109/TVCG.2007.70568]

[71] Plumlee MD, Ware C. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. ACM Trans. on Computer Human Interaction, 2006,13(2):179−209. [doi: 10.1145/1165734.1165736]

[72] Chen Y, Zhang XY, Feng YC, Liang J, Chen HQ. Sunburst with ordered nodes based on hierarchical clustering: A visual analyzing method for associated hierarchical pesticide residue data. Journal of Visualization, 2015,18(2):237−254. [doi: 10.1077/s12650-014-0269-3]

附中文参考文献:

[1] 程学旗,靳小龙,王元卓,郭嘉丰,张铁赢,李国杰.大数据系统和分析技术综述.软件学报,2014,25(9):1889−1908. http://www.jos.org.cn/1000-9825/4674.html [doi: 10.13328/j.cnki.jos.004674]

[28] 肖卫东,孙扬,赵翔,周城,封孝生.层次信息可视化技术研究综述.小型微型计算机系统,2011,32(1):137−146.

[31] 张昕,袁晓如.树图可视化.计算机辅助设计与图形学学报,2012,24(9):1113−1124.

[62] 袁晓如,张昕,肖何,郭翰琦.可视化研究前沿及展望.科研信息化技术与应用,2011,2(4):3−13.

[63] 任磊,杜一,马帅,张小龙,戴国忠.大数据可视分析综述.软件学报,2014,25(9):1909−1936. http://www.jos.org.cn/1000-9825/4645.html [doi: 10.13328/j.cnki.jos.004645]

[69] 任磊,王威信,滕东兴,马翠霞,戴国忠,王宏安.海量层次信息的 Focus+Context 交互式可视化技术.软件学报,2008,19(11):3073−3082. http://www.jos.org.cn/1000-9825/19/3073.htm

李彦龙(1989－),男,河北石家庄人,硕士,河北石家庄人,主要研究领域为层次数据可视化,可视分析.

李国强(1979－),男,博士,副教授,CCF 专业会员,主要研究领域为形式化验证,计算机理论,可计算学习理论.

董笑菊(1975－),女,博士,副教授,CCF 高级会员,主要研究领域为并发理论,形式化方法,可视化,可视分析.

© 中国科学院软件研究所 http://www.jos.org.cn