

# TVBRT:一种基于 Radial Tree 的具有度量属性的多变元时态数据可视化方法

孙宁伟<sup>1</sup> 赵瑜<sup>1</sup> 刘勇<sup>1</sup> 刘海峰<sup>1</sup> 肖卫东<sup>2</sup> 张翀<sup>2</sup>

(中国人民解放军 91655 部队 北京 100036)<sup>1</sup> (国防科技大学信息系统工程重点实验室 长沙 410073)<sup>2</sup>

**摘要** 多变元时态数据大量存在于社会科学、环境监测、金融经济、医疗卫生及地理信息等领域。对多变元时态数据集进行深入分析,创新性地经典的层次数据可视化方法 Radial Tree 加以改进,提出集成度量属性的 Radial Tree 布局算法(LAMPRT)、基于 Radial Tree 的具有度量属性的多变元时态数据可视化布局算法(LOVEBRT),结合相应的人机交互策略,形成可视化方法 TVBRT。案例研究证明了该方法的有效性,并表明它能展示数据集中的细节内容。TVBRT 方法侧重于对数据进行层次分析,适合于展示对具体度量值更为关心的数据集。

**关键词** 信息可视化,时态,多变元数据,度量属性,Radial Tree

**中图分类号** TP391 **文献标识码** A

## TVBRT: A Time-varying Multivariate Data Visualization Method Based on Radial Tree

SUN Ning-wei<sup>1</sup> ZHAO Yu<sup>1</sup> LIU Yong<sup>1</sup> LIU Hai-feng<sup>1</sup> XIAO Wei-dong<sup>2</sup> ZHANG Chong<sup>2</sup>

(Unit 91655, People's Liberation Army, Beijing 100036, China)<sup>1</sup>

(Science and Technology on Information Systems Engineering LAB, National University of Defense Technology, Changsha 410073, China)<sup>2</sup>

**Abstract** The areas of social science, environmental monitoring, financial and economic, health and geographic information have generated a large amount of time-varying multivariate data. With a depth analysis of the time-varying multivariate datasets, we proposed a layout algorithm for Radial Tree integrated metric properties called LAMPRT to demonstrate metric properties. And then, we proposed another layout algorithm called LOVEBRT to exhibit the time-varying feature. These two algorithms and some human-computer interaction techniques both form the visualization method TVBRT. Case studies demonstrate the effectiveness of this method, showing that the method is more effective in exhibiting more details of the datasets. TVBRT outperforms over other methods in exhibiting data in a hierarchical and more accurate way.

**Keywords** Information visualization, Time-varying, Multivariate data, Metric properties, Radial tree

## 1 引言

对多变元时态数据的可视化研究工作已经成为信息可视化领域的一个新兴热点,受到研究人员的高度重视,2012年10月份召开的信息可视化领域顶级会议 IEEE InfoVis 2012 中有专门的相关议题。具有度量属性的多变元时态数据的分析任务往往是关注多变元数据的度量值随时间的变化趋势情况,若对该变化趋势情况进行可视化,则需要在视图中同时展现分类型和数值型变元、同时展现多变元和时态特性。

文献[27]给出了多变元时态数据的基本概念,并进行了相关研究;文献[1]的方法主要是将多变元可视化视图与时态可视化视图拼凑在一起。

Jimmy Johansson 等人通过对平行坐标系进行改进开展多变元时态数据集可视化研究<sup>[5]</sup>,提出一种使用多边形形式渲染不同时间段之间数据变化趋势的概念;并在其博士论文

中对多变元时态数据集可视化技术进行了深入讨论<sup>[6]</sup>,使用三维多关系平行坐标系技术对多变元时态数据进行可视化,对标准二维平行坐标系和三维多关系平行坐标系分析数据的情况进行评价,在平行坐标系表示方法中使用高精密度图(high precision density maps)表现结构方面的信息,使用转移函数(transfer functions)揭示大规模多变元数据集在平行坐标系表示方法中的复杂结构,用时态平行坐标系技术实现对大规模多变元时态数据的交互式分析。这类方法在一定程度上克服了两幅视图之间映射造成的认知负担,但渲染过程往往需要较为复杂的计算,算法效率不高。

Teng-Yok Lee 等人基于变元值随时间的变化情况,以及不同空间区域和时间间隔内各变元值变化情况之间的相互关系识别出变元之间的趋势关系,提出了一种可视化多变元时态数据集的分析方法<sup>[7]</sup>,其不但支持对时态趋势的发现和可视化,而且能够对多个变元之间的趋势进行时空关系的建模;

收稿日期:2013-05-19 返修日期:2013-10-14

孙宁伟(1987—),男,硕士,助理工程师,主要研究方向为信息可视化、数据挖掘、计算机网络,E-mail:sunningwei@sina.cn;赵瑜(1974—),女,工程师,主要研究方向为计算机网络、网络管理;刘勇(1976—),男,工程师,主要研究方向为计算机网络、数据中心;刘海峰(1973—),女,高级工程师,主要研究方向为计算机网络;肖卫东(1968—),男,教授,博士生导师,主要研究方向为信息资源管理、指挥信息系统;张翀(1981—),男,博士,讲师,主要研究方向为信息资源管理。

Cheng-Kai Chen 等人的工作<sup>[8]</sup>针对科学计算中的气象数据集展开,使用采样数据的相互关系来近似整体数据集的相互关系,大大减小了需要处理的数据量;Conglei Shi 等人以搜索引擎检索词的数量和相对重要性为例<sup>[9]</sup>,基于河流模型(ThemeRiver)提出了一种新型可视化原型系统 RankExplorer,它能够挖掘排序变化情况的隐含模式。这类研究能够减少需要处理的数据项数量,但只能对感兴趣的趋势进行识别,不利于挖掘出事先不知道的、隐含的趋势,有可能造成信息损失。

具有度量属性的多变元时态数据属于“混合数据”的范畴,现有的混合数据可视化方法多基于已有的分类型或数值型数据可视化方法,采取一定的数据转换技术对多变元的数据类型进行统一<sup>[10]</sup>。在数据转换的过程中不可避免地会导致信息的损失,且不利于在原始数据集上对发现的模式做出合理解释。

## 2 相关工作

### 2.1 层次化方法分析多变元时态数据集

具有度量属性的多变元数据集中每一变元均以分类型数据占主体,在不考虑度量值的情况下该数据集实际上是分类型数据集。事实上,分类型数据集在某些变元的取值确定之后,从该变元出发继续分析,数据集会呈现出一定的层次性,如图1以销售数据为例分析分类型数据集的层次性。在图(a)中按照“地域”-“货物类别”-“时间”的顺序进行分析,从“地域”变元出发,选定北京地区之后,进一步分析“货物类别”、“时间”,便可得到图(b),从而形成一个层次结构的视图,当“地域”选择其他取值或者分析顺序发生变化时,会形成不同的层次结构视图。

具有度量属性的多变元时态数据的分析任务除了包括分析某一变元的时态特性和多个变元的多变元分布特性、对比多个变元的时态特性之外,还应该包括从不同的变元出发,按照不同的变元顺序分析数据集中的层次特征,以及分析不同的层次视图中各变元随时间的变化情况,以充分挖掘数据集中隐含的知识。基于此认识,提出基于层次可视化方法的具有度量属性的多变元时态数据可视化方法。

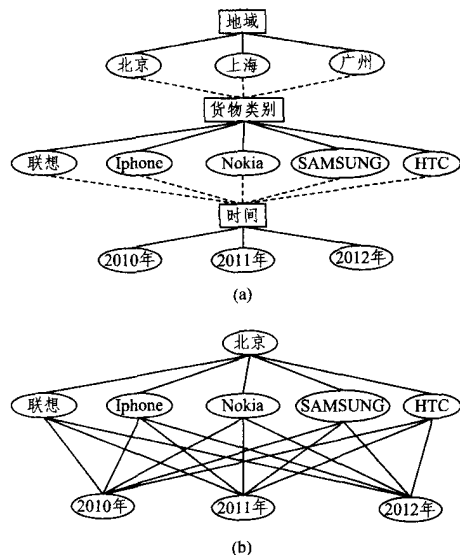


图1 销售数据层次化分析案例

### 2.2 Radial Tree

Radial Tree 方法采用径向布局方法<sup>[13]</sup>,根节点绘制在圆心处,其余节点按照所属层次的不同被放置在不同半径的同心圆上,节点到圆心的距离对应于该节点的深度,每棵树的子节点排布在同心圆的放射状楔形圆弧上,每一个双亲节点都有各自的楔形范围,从而避免了子树的重叠。孩子节点排列在距离根节点最近的一个圆环上,孙子在次近的圆环上。Radial Tree 方法提供良好的人机交互方法,支持变换不同的变元分析顺序,分析方式灵活,如果能够在视图中集成数据集中的度量属性和时态特性,则既可以展现数据集中的多变元特性和时态特性,又不必进行任何的数据类型的转换,还能完成多变元时态数据层次性分析的任务。

文献<sup>[15]</sup>基于 Radial Tree 提出“parent-centered Radial Tree”布局方式,即视图中每一个节点均以其父节点重新构建坐标系,父节点作为圆心,子节点排列在以父节点为圆心的楔形圆弧上,这种布局方式可以有效避免视图出现过多的混乱,明确展现出数据集中的层次关系。

## 3 TVBRT: 基于 Radial Tree 的具有度量属性的多变元时态数据可视化方法

### 3.1 定义

根据文献<sup>[11]</sup>对多维数据的定义,给出具有度量属性的多变元数据集定义,如下:

定义1(具有度量属性的多变元数据集) 设  $G(F) = \{F^1, F^2, \dots, F^m\}$  为给定的  $k(k > 0)$  变元对象集合,其中,  $m$  为集合基数,即多变元对象的数量,  $F^i = F^i(c_1^i, c_2^i, \dots, c_{k-1}^i, q^i)$  ( $q^i \in R, q^i \geq 0$ ) 代表集合中的一个  $k$  变元对象,  $c_1^i, c_2^i, \dots, c_{k-1}^i$  表示每个对象的  $(k-1)$  变元属性,且均为分类型数据,  $q^i$  表示由  $c_1^i, c_2^i, \dots, c_{k-1}^i$  确定的数据项的度量值(通常为概率或数量),称  $G(F)$  为具有度量属性的多变元数据集。对于多变元属性数据表  $G(F)$ ,  $F^i$  表示一条记录,  $q^i$  称为该条记录的度量。

最常见的具有度量属性的多变元数据集是销售数据,销售数据包括地域、货物类别、时间和销售数量等维度(变元),当固定了地域、货物类别、时间之后,销售数量即成为所确定的数据项的度量。

定义2(度量变元) 对具有度量属性的多变元数据集  $G(F)$ ,定义其  $k$  个变元,分别表示为:  $\{V_1, V_2, \dots, V_{k-1}, Q\}$ ,其中  $V_i(1 \leq i \leq k-1)$  均为分类型变元,  $Q$  为数值型变元,且变元  $Q$  的任一取值均为非负实数,称  $Q$  为  $G(F)$  的度量变元。

定义3(分类型的度量) 考虑具有度量属性的多变元数据集  $G(F)$  中的变元  $V_l(1 \leq l \leq k-1)$ ,设  $\{v_1, v_2, \dots, v_n\}$  是  $V_l$  的  $n$  个分类值,那么任意的分类值  $v_i(1 \leq i \leq n)$  在整个数据集  $G(F)$  中的度量可表示为  $Q(v_i) = \sum_{j=1}^m q^j |_{V_l=v_i}$ ,即  $Q(v_i)$  表示变元  $V_l$  取分类值  $v_i$  的所有度量值之和,称  $Q(v_i)$  为分类值  $v_i(1 \leq i \leq n)$  的度量。

定义4(具有度量属性的多变元时态数据集) 对具有度量属性的多变元数据集  $G(F)$ ,定义其  $k$  个变元,分别表示为:

表1 LAMPRT 算法

$\{V_1, V_2, \dots, V_{k-1}, Q\}$ , 若某一变元  $V_i (1 \leq i \leq k-1)$  的取值为若干个时间节点, 则整个数据集具有一定的时态特性, 称这样的数据集  $G(F)$  为具有度量属性的多变元时态数据集。

3.2 集成度量属性的 Radial Tree 布局算法(LAMPRT)

Radial Tree 视图中的节点均使用同一种方式表示, 节点不能展示太多的信息, 若对具有度量属性的多变元时态数据集进行分析, 则必须将数据集中的变元以及变元的各个分类值同时在可视化视图中进行展现, 因此需要对 Radial Tree 视图中的节点进行改进。

借鉴相关研究<sup>[15]</sup>, 使用正比于变元和分类值度量值大小的节点对数据集中的相关信息进行量化。使用矩形表示变元节点, 圆形表示分类值节点, 并将变元抽象为父节点, 变元的各分类值抽象为该变元的子节点, 将变元和该变元各分类值着相同颜色, 以保持认知上的统一性, 避免理解上可能出现的混乱。为在视图中明确展示出变元(父节点)和变元分类值(子节点)之间的父子关系, 基于 parent-centered Radial Tree 布局方法提出本文布局算法。

图2是本小节布局算法的示意图, 图中  $V_1, V_2$  表示两个变元,  $a, b, c, d, e, f$  分别表示分类值, 各变元、分类值节点的大小正比于相关的度量值。将根节点布局在  $C_0$  处, 首先根据第一层变元相互之间不出现交叠的原则, 计算出第一层圆的半径  $R$ 。然后将其中两个变元根据相关排列方法分别布局在  $C_1, C_2$  处, 计算出以两个变元为根节点的第二层圆的半径分别为  $R_1, R_2$ , 圆  $C_1$ 、圆  $C_2$  相交于  $A, F$  两点, 根据 parent-centered 布局方法, 分别计算出以  $C_1, C_2$  为根节点的子节点(即变元的分类值)可以布局的范围, 分别为小于半圆的弧  $AB$  和  $DE$ , 并根据一定的排列规则将各分类值在圆弧上依次排列。重复上述过程, 即可以完成整个布局过程。结合上述描述内容, 对 parent-centered Radial Tree 布局算法进行改进, 基于极坐标的表示形式, 提出集成度量属性的 Radial Tree 布局算法(Layout Algorithm for visualizing multivariate data with Measures based on Parent-centered Radial Tree, LAMPRT), 如表1所列。

算法1 LAMPRT 算法

- Step 1 计算变元和分类值的度量值, 并进行归一化处理, 用归一化之后的数值作为变元和分类值节点在视图中的面积。设变元面积大小依次为  $\{V_1, V_2, \dots, V_n\}$ 、各变元对应的分类值面积依次为  $\{c_1^1, c_2^1, \dots, c_{m_1}^1, c_1^2, c_2^2, \dots, c_{m_2}^2, \dots, c_{m_n}^n\}$ , 其中  $n$  表示变元个数,  $m_i$  表示第  $i$  个变元的分类值的个数。
- Step 2 在视图中心布局圆心节点  $C_0$ 。
- Step 3 指定第一层圆的半径  $R$ , 满足  $R \geq 2\pi / \sum_{i=1}^n V_i$ , 并依次将各变元矩形节点在圆弧上进行排列, 位置分别记为  $\{C_1, C_2, \dots, C_n\}$ 。
- Step 4 设圆心节点  $C_i$  的极坐标位置为  $(0, R)$ , 根据用户指定的该层节点允许存在的角度范围  $\Omega_i (0 < \Omega_i \leq 180)$ , 设以  $C_i$  为父节点的分类值可以布局的角度范围为  $(\alpha, \beta)$ , 则  $(\alpha, \beta)$  的极坐标表示形式中的角度值可以通过如下公式计算:  

$$\alpha = \theta_i + (360 - \frac{1}{2} \Omega_i)$$

$$\beta = \alpha + \Omega_i$$
- Step 5 采用如下方式计算以  $C_i$  为圆心的该层圆的半径  $R_i$  大小:  
 if 以  $C_i$  为父节点的分类值个数  $= 1$  then  

$$R_i = \frac{1}{2} R$$
 else  

$$R_i = \frac{360}{\Omega_i} \frac{1}{2\pi} \sum_{j=1}^{m_i} c_j^i + r_i / r_i$$
 为用户定义的半径修正值  
 end if.
- Step 6 以  $C_i$  为圆心、 $R_i$  为半径, 绘制  $(\alpha, \beta)$  范围内的圆弧, 并在圆弧上按照分类值所占的面积大小进行节点布局。
- Step 7 重复 Step 4—Step 6, 直至完成所有以  $\{C_1, C_2, \dots, C_n\}$  为圆心的分类值的布局。
- Step 8 如果要以某分类值为圆心进行进一步层次分析, 则以该分类值为圆心, 重复 Step 4—Step 6, 对各变元依次进行排列, 形成层次分析视图。

LAMPRT 算法在节点布局的过程中充分考虑节点大小的因素, 在计算第二层圆时, 通过分类值面积大小确定所绘圆弧的半径, 使每个变元或分类值在视图中所占的区域大小正比于其度量值大小, 以便直观比较各个分类值之间的大小关系。当变元或分类值数量较多时, 如果将所有元素均在视图中进行显示, 可能会导致视图非常繁杂, 在具体可视化实现过程中采取的策略是: 首次渲染时只显示一部分变元或分类值, 提供人机交互方法, 使用户根据自己感兴趣的元素查看未显示的部分, 例如在图2中, 用户可点击节点“more”显示变元的全部分类值。如果需要在两层视图的基础上进行进一步的层次分析, 则可以选择关心的分类值作为圆心节点, 算法可以按照相同的方式构造出多层视图。

算法中需要用户指定的参数包括: 第一层圆的半径  $R \geq 2\pi / \sum_{i=1}^n V_i$ 、第二层节点允许存在的角度范围  $\Omega_i (0 < \Omega_i \leq 180)$ , 以及第二层圆弧半径的修正值  $r_i$ 。这些参数需要在具体的程序实现过程中根据数据集和可视化效果情况进行指定。总体来说, 该算法的时间空间复杂度均为  $O(n)$ 。

3.3 基于 Radial Tree 的具有度量属性的多变元时态数据可视化布局算法(LOVEBRT)

为展示数据集中的时态变化特性, 本小节在集成度量属性的 Radial Tree 的基础上进行扩展, 借鉴经典的表现数据时态变化趋势的折线图方式, 按照要比较的时间段的个数分别生成多个 Radial Tree 视图, 并采取一定的布局算法, 使关注的变元和分类值按照度量值的大小排列在不同的折线上, 以实现不同变元、同一变元不同分类值的时态变化情况的分析。

多变元时态数据的时态分析任务包括对单个变元及其各

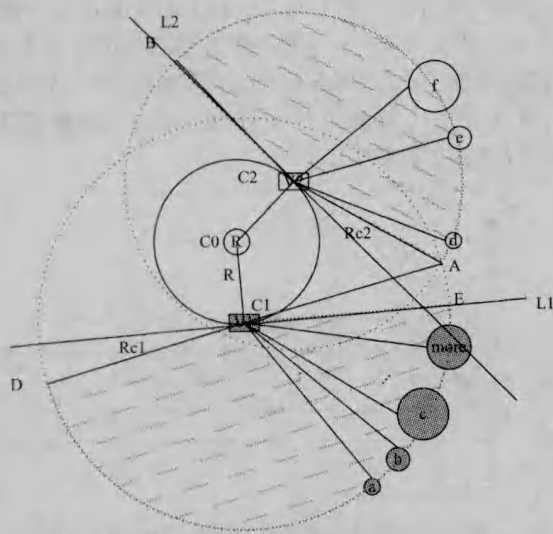


图2 LAMPRT 布局算法示意图

分类值进行时态变化趋势分析,以及对多个变元联合进行时态趋势分析两个方面,分别用图 3、图 4 进行示意,横轴表示时间,纵轴表示度量值的大小。

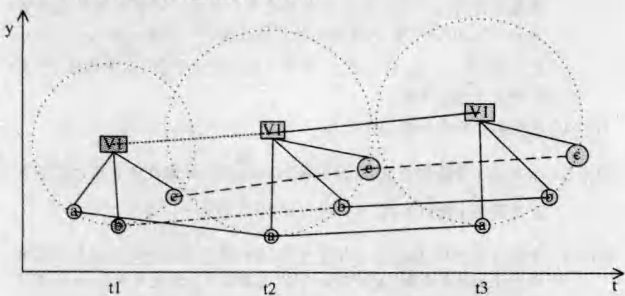


图 3 单个变元分析时态 Radial Tree 示意图

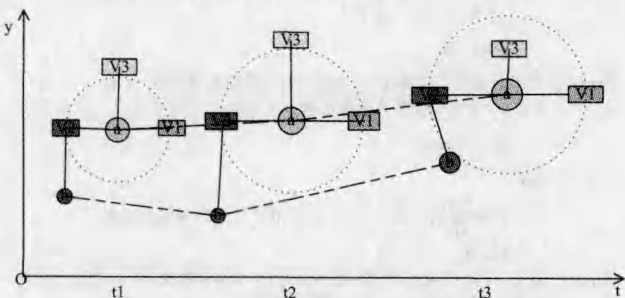


图 4 多个变元分析时态 Radial Tree 示意图

图 3 展示单个变元及其各分类值的时态趋势,待分析的变元分别排列在各个时刻视图的中央,其分类值依次排在以变元为圆心的圆弧上,并且各个变元和分类值在纵轴上的投影对应于度量值的大小。分别用 4 种不同的虚线展示了各个变元和分类值的度量值大小变化趋势。从该图中可以明显看出,  $V_1$  变元的取值在整个时刻内不断增大;分类值  $a$  在  $t_1$  到  $t_2$  时刻内减小,在  $t_2$  到  $t_3$  时刻内增大;分类值  $b$  和  $c$  在整个时刻内不断增大。进一步可以看出分类值  $c$  的增大趋势最为明显,度量值最大,因此可以猜测分类值  $c$  在变元  $V_1$  中的重要性最为突出。

特别需要注意的是,对多个变元进行基于时态 Radial Tree 分析的具体含义是,对某个变元的某一分类值进一步展开视图,形成层次分析过程,将从该分类值出发的第二个变元的某一分类值与该分类值进行对比分析,其实质是对多个变元的分类值的一种层次性分析。进行多个变元时态分析时,为使视图更加简洁明显,默认情况下不显示所有变元的分类值,而只显示将两个变元联系在一起的分类值,如图 4 中节点  $a$  即为将要分析的变元  $V_2$  和  $V_3$  联系到一起的分类值。用户可根据需要对默认显示状态进行修改。

图 4 展示多个变元的时态分析,各个变元分别用不同的颜色着色( $V_1$  为红色、 $V_2$  为绿色、 $V_3$  为黄色),需要展开分析的两个变元分别是  $V_2$ 、 $V_3$ ,两个变元通过分类值  $a$  联系在一起,并选取分类值  $b$  作为  $V_2$  变元的代表分类值。从该图中可以明显看出,在  $t_1$  到  $t_2$  时刻内  $a$  的度量值在增加,而  $b$  的度量值在减小;在  $t_2$  到  $t_3$  时刻内  $a$ 、 $b$  的度量值均在增加, $b$  的增加幅度更加明显。

从上述图中可以看出,时态 Radial Tree 的绘制首先需要明确要分析的时间段,然后进行布局。整个视图的绘制顺序可以分为以下几个阶段:

(1) 确定第一个 Radial Tree 的位置,并根据分析任务确定要展示的元素,绘制第一时刻的视图;

(2) 按照一定原则制定相关布局策略,绘制第二时刻视图;

(3) 对前两个时刻的绘图方式进行重复迭代,直至全部时间段的视图绘制完成。

为减少初始视图中的混乱,需要在初始视图中隐藏不关注的内容。在默认情况下,第(1)阶段提供的视图分为以下 3 种情况:

(1.1) 如果进行单个变元的时态分析, Radial Tree 视图展示该变元及其所包含的各个分类值;

(1.2) 如果进行某一分类值的时态特性的分析, Radial Tree 视图展示该分类值、分类值所属变元、所属变元其他分类值;

(1.3) 如果进行多个变元联合时态分析, Radial Tree 视图展示要分析的多个变元及多个变元的代表分类值。

用户可以根据自己的分析需求修改上述默认值,使初始绘制的视图包含更多的元素,同时由于上述(1.1)和(1.2)两种情况均展示变元及其各个分类值,在进一步分析时将其归类为一种情况。

第(2)阶段在整个过程中最为关键,如何在已有第一时刻视图的基础上,对第二时刻的 Radial Tree 视图进行布局是要解决的最主要的问题,此处指定的布局原则与前述保持一致,即尽量减少视图中出现节点重叠现象。

图 4 实际上是图 3 按照变元的某一分类值进行层次展开之后获得的视图,因此整个布局过程中需要解决的问题实质是首先完成单个变元分析的视图构建,然后根据分析需要对视图进行层次展开,层次展开之后的视图即为多个变元分析的视图。下面重点对单个变元分析时第二时刻图的绘制方法以及具体分类值层次展开的方法进行说明。

图 5 展示单个变元分析 Radial Tree 的布局示意图。绘制  $t_1$  时刻 Radial Tree 的过程大致是:

(1) 以视图的左下角为坐标原点,绘制时间-度量坐标轴  $yot$ 。

(2) 隐藏原视图中不需要展示的元素。

(3) 设需要展现变元的度量值大小为  $y_0$ , 该变元各分类值的度量值分别为  $\{V_1, V_2, \dots, V_n\}$ , 该变元与其分类值组成的 Radial Tree 的半径  $R_1 = y_0 - \min(V_1, V_2, \dots, V_n)$ 。假设该变元度量值最大的分类值对应的圆形节点的半径为  $r_1$ , 则  $r_1 = \sqrt{\max(V_1, V_2, \dots, V_n) / \pi}$ , 以  $y$  轴上的取值  $y_0$  处为圆心(记作  $C_0$ )、以  $R_0 = R_1 + r_1$  为半径画圆, 该圆与过  $y_0$  垂直于  $y$  轴的射线相交于点  $C_1$ , 则点  $C_1$  即为需要展现的变元的位置。

(4) 将变元  $V_1$  移动到点  $C_1$  处, 变元的各分类值根据度量值的大小排列在圆弧上。

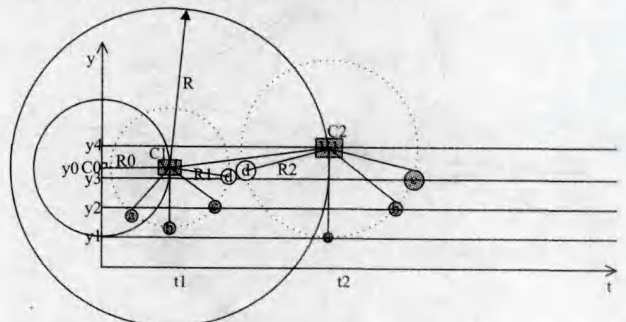


图 5 单个变元分析第二时刻时态 Radial Tree 布局示意图

在  $t_1$  时刻视图的基础上绘制  $t_2$  时刻的 Radial Tree 图的

过程大致是:

(1)分别从  $y$  轴引出 4 条平行线  $y_1, y_2, y_3, y_4$  ( $y_1 < y_2 < y_3 < y_4$ ), 其坐标值分别对应分类值和各变元的度量值的大小, 其中  $y_4$  为变元的度量值, 则  $t_2$  时刻 Radial Tree 的圆半径可以表示为  $R_2 = y_4 - y_1$ ;

(2)假设极端情况下, 两个变元的最大的分类值恰好相切, 分别记度量值最大的圆的半径为  $r_1, r_2$ , 则  $C_1 C_2$  之间的距离需要满足  $|C_1 C_2| \geq R_1 + R_2 + r_1 + r_2$ ;

(3)以  $C_1$  为圆心、 $R=R_1+R_2+r_1+r_2+r_0$  ( $r_0$  为用户指定的半径修正值) 为半径画圆, 与  $y_4$  直线的交点即为  $t_2$  时刻径向图的圆心  $C_2$ , 然后即可根据度量值将各分类值在圆弧上进行布局, 绘制出  $t_2$  时刻的 Radial Tree。

此时若需进行多个变元的分析, 则需要对分类值进行层次展开, 首先需要从该分类值开始选择要展开的变元, 由于变元的度量值与分类值的度量值实际上大小相等, 因此变元矩形节点与该分类值圆形节点位于同一水平轴线上。参考上述单个变元分析 Radial Tree 的布局过程, 即可进一步展开该变元的各个分类值, 从而形成支持多个变元联合分析的层次视图。

在具有度量属性的 Radial Tree 布局算法 LAMPRT 完成的可视化视图的基础上, 给出基于 Radial Tree 的具有度量属性的多变元时态数据可视化布局算法 (Layout algorithm for visualizing time-Varying multivariate data Based on Radial Tree, LOVEBRT), 如表 2 所列。

表 2 LOVEBRT 算法

算法 2	LOVEBRT 算法
Step 1	确定要展示时态视图的时间节点, 设为 $\{t_1, t_2, \dots, t_n\}$ 。
Step 2	调用 LAMPRT, 完成初始 Radial Tree 的绘制。
Step 3	以视图中最左下角为原点, 绘制二维平面坐标轴 $yot$ , 横轴表示时间, 纵轴表示度量值 (归一化)。
Step 4	$r_1$ 表示 $t_1$ 时刻变元的分类值中度量值最大的圆形节点的半径, 若变元的度量值大小为 $y_0$ , 该变元各分类值的度量值分别为 $\{V_1, V_2, \dots, V_n\}$ 。 记 $V_{\max} = \max(V_1, V_2, \dots, V_n), V_{\min} = \min(V_1, V_2, \dots, V_n)$ , 则有: $r_1 = \sqrt{V_{\max}/\pi}, R_1 = y_0 - V_{\min}$ 。
Step 5	选择要进行时态展示的节点, 根据选择的节点的类型, 对已有 Radial Tree 布局视图进行如下调整: if 节点类型是变元 then 保留视图中选中节点及其子节点; else 保留视图中选中节点及其父节点。
Step 6	过 $y_0$ 作垂直于 $y$ 轴的直线 $l_0$ , $y$ 轴与 $l_0$ 交于点 $C_0$ , 以 $C_0$ 为圆心、 $R_0 = R_1 + r_1$ 为半径作圆, 该圆与直线 $l_0$ 相交于 $C_1$ , 将现有视图中的矩形节点调整到点 $C_1$ , 各分类值按照度量值在圆弧上布局。
Step 7	设 $t_2$ 时刻需要展示的各子节点元素的度量值为 $\{cy_1, cy_2, \dots, cy_n\}$ , 父节点的度量值分别为 $y_{t_2,0}$ 。 记 $y_{\max} = \max(cy_1, cy_2, \dots, cy_n), y_{\min} = \min(cy_1, cy_2, \dots, cy_n)$ , 则 Radial Tree 的圆半径为: $R_2 = y_{t_2,0} - y_{\min}$ 。
Step 8	设 $r_2$ 为 $t_2$ 时刻 Radial Tree 中分类值度量值最大的圆形节点的半径, 则 $r_2 = \sqrt{y_{\max}/\pi}$ 。 以 $C_1$ 为圆心、 $R=R_1+R_2+r_1+r_2+r_0$ ( $r_0$ 为用户指定的半径修正值) 为半径画圆, 与 $y_{t_2,0}$ 直线的交点即为 $t_2$ 时刻径向图的圆心 $C_2$ 。
Step 9	以 $C_2$ 为圆心、 $R_2$ 为半径构造 Radial Tree, 根据度量值在 $y$ 轴上的映射对各节点进行布局, 完成 $t_2$ 时刻 Radial Tree 的绘制。
Step 10	重复迭代 Step 7—Step 9, 直至全部时间段的视图绘制完成。
Step 11	若需进行多个变元分析, 则首先选择某一个待分析变元的一个分类值, 然后展开该分类值另外要分析的变元, 分别与该分类值排列在同一水平轴上。然后选取展开的变元重复迭代 Step 4—Step 10, 完成层次结构的视图。

从上述算法描述中可以看出, 算法的重点在于确定后续时间段的 Radial Tree 的圆心位置, 其基本思路是通过一个大圆与相关直线的交点来确定圆心位置, 且只需遍历变元的各个分类值, 因此计算过程并不复杂, 时间空间复杂度均为  $O(n)$ 。根据时间节点数目的不同, 需要迭代计算构造出多个 Radial Tree, 其空间复杂度与时间段数量和分类值个数成正比, 因此为  $O(n)$ 。在上述算法中尽量控制了需要同时显示的节点数量, 以使视图的渲染和运行占用尽量少的计算资源。

### 3.4 人机交互策略

本小节介绍 TVBRT 可视化方法的人机交互策略。

Step 1 调用 LAMPRT, 在 Radial Tree 视图上集成变元和变元各分类值的度量值, 并按照每个变元每次最多显示若干个 (例如 10 个) 分类值的方式构造 Radial Tree 视图。

Step 2 为各变元及各分类值提供合理的人机交互操作方式。

Step 2.1 某具体变元的交互操作主要包括:

(1)鼠标单击代表变元的矩形框, 变元在视图中居中, 视图重新布局;

(2)鼠标双击代表变元的矩形框, 变元被设置为 Radial Tree 的根节点, 视图重新布局, 该变元成为层次分析的起点;

(3)变元分类值超过规定值时, 出现一个与变元相连的圆形节点“more”, 点击该节点, 则显示该变元所有的分类值;

(4)在代表变元的矩形框上点击鼠标右键, 出现“时态分析”菜单, 点击该菜单, 则转入 Step 3, 生成变元的时态趋势图。

Step 2.2 某具体分类值的交互操作主要包括:

(1)鼠标单击代表分类值的圆形节点, 分类值在视图中居中, 视图重新布局;

(2)鼠标双击代表分类值的圆形节点, 分类值被设置为 Radial Tree 的根节点, 视图重新布局, 该节点作为层次分析的起点, 同时出现代表其他所有变元的矩形框与该分类值相连, 这些矩形框均支持 Step 2.1 中所有的交互式操作;

(3)在代表分类值的圆形节点上点击鼠标右键, 出现“时态分析”菜单, 点击该菜单, 则转入 Step 3, 生成该分类值及该分类值所属变元的时态趋势图。

Step 2.3 整个视图的交互操作主要包括:

(1)提供时态选择窗口, 用户既可以选中特定的时刻, 并生成当前时刻的视图, 以便进行某一具体时间段的多变元分析, 又可以选中多个时刻, 转入 Step 3, 进行时态变化分析;

(2)提供变元选择窗口, 用户可以选择感兴趣的若干个变元, 生成所选变元的时态趋势图, 以便进行多变元联合时态趋势分析。

Step 3 调用 LOVEBRT, 对 Radial Tree 视图进行调整, 以实现具有度量属性的多变元时态数据的基于 Radial Tree 方式的可视化分析, 当鼠标悬停在布局完成的视图中的某一节点上时, 按照度量值的大小绘制相关的折线图来表现度量值的时态变化情况。同时视图仍然提供 Step 2 中的交互方式, 并实现多个 Radial Tree 之间的联动。

## 4 实验结果

首先对数据集和实验环境进行描述和说明; 然后分别验证 LAMPRT 算法、LOVEBRT 算法的有效性, 结合 TVBRT

可视化效果进行分析,挖掘实验数据集中的隐含知识;最后对可视化方法的特点进行说明。

#### 4.1 数据集描述与实验环境说明

本文以院校招生数据集为例对提出的可视化方法进行案例研究,该数据集积累了近年来大量的招生计划数据,包括年度、招生院校、招生专业、学制、学历、性别、文/理科、划线类别、学生生源地等 11 个分类型变元(维度),每一条记录还对应一个具体的度量值变元即招生数量,该变元取值类型为数值型,整个数据集记录约有 40000 条,包含的时间节点(年度)为 8 个。

该数据集符合本文给出的具有度量属性的多变元时态数据集的定义,招生数量是数据集中的唯一数值型变元即度量变元,除招生数量外的变元均为分类型,数据集中包含若干年度的记录,可以根据年度这一变元进行时态分析。开展案例研究的硬件环境为: Intel(R) Core(TM)2 Duo CPU P8600 2.40GHz, 2.93GB 内存物理地址, ATI Mobility Radeon HD 3400 Series 显卡, 320GB 硬盘; 软件环境为: Microsoft Windows XP Professional SP3, Eclipse, MyEclipse 6.5, JDK1.6, Oracle 9i; 相关的开源软件包为: Prefuse<sup>[25,26]</sup>。

#### 4.2 可视化效果

本小节可视化视图中,矩形表示变元,圆形表示变元具体的分类值;不同颜色代表不同的变元,同一种颜色表示同一种变元。

##### 4.2.1 LAMPRT 算法的案例验证

为验证在 Radial Tree 中集成度量属性表现数据集度量属性的有效性,选取数据集中的若干变元和分类值,使用 LAMPRT 算法进行可视化布局,对形成的可视化视图进行讨论和分析。

图 6 以 2010 年招生计划数据为例对其中的 8 个变元进行了展示,分别对每个变元设置了最多允许显示的分类值的个数,使得视图尽量美观。例如,左上角的橘黄色矩形表示“生源地”变元,连接到该矩形的橘黄色圆形分别代表“生源地”的分类值:安徽、福建、北京、广州、甘肃等。图中变元按照顺时针方向,依次为:生源地、学历、学制、性别、培养目标、专业、院校、专业类别。同时通过人机交互方式,将总节点调整到视图中间,各变元节点调整到视图第二层,变元节点的分类值位于最外层。

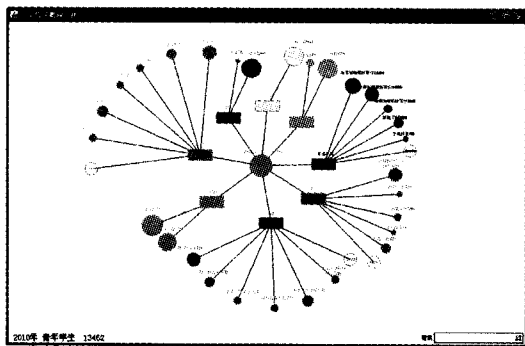


图 6 LAMPRT 算法可视化效果图

如果想查看变元所有的分类值,需要点击“more”节点;每个变元的各个分类值圆节点的大小在本变元内部互成比例。视图左下角提示当前鼠标悬停节点的详细信息,视图右下角提供了搜索输入框,用户可以输入相关关键词,视图中满

足关键词的节点将会用高亮色进行显示。从图中可以非常直观地看出,院校、专业、培养目标、生源地 4 个变元的分类值取值较多,而性别、学制、学历、专业类别等变元的取值较少。

##### 4.2.2 LOVEBRT 算法的案例验证

为验证使用多个 Radial Tree 表现数据集时态特性的有效性,分别选取 3 个不同时间点的数据进行可视化效果展示,并对可视化视图进行分析,以有效展现出数据集中有关变元和分类值度量值的变化情况,挖掘数据中隐含的时态特性。

图 7 以 2010—2012 年国防科大计算机科学与技术、仿真工程、机械工程及其自动化、应用物理、指挥自动化工程等 5 个专业的招生情况进行了 LOVEBRT 算法的验证。图中绿色矩形表示变元“专业”,绿色圆形表示上述 5 个专业。

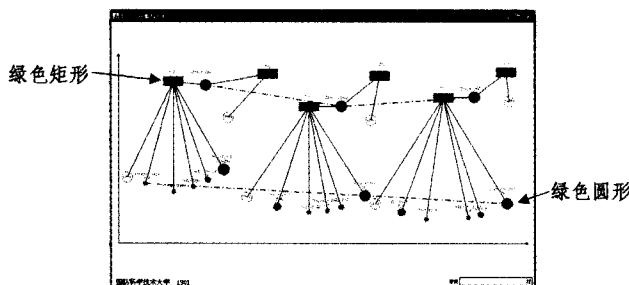


图 7 LOVEBRT 算法可视化效果图

在视图中可以看出 3 年之间国防科大的招生数量在变化,2010 年到 2011 年之间减小比较明显,2011 年到 2012 年之间出现增加,所研究的 5 个专业的招生数量都在逐年减少,由此印证了近年来有关院校培养模式的调整;进一步还可以看出从 2010 年到 2011 年两年之间的招生数量变化非常大,这实际上是由于从 2011 年开始在招生计划中将指挥类专业进行了单列,导致相关的技术类专业的招生数量在统计上出现了较大程度的减少。图中各个变元分别用不同的颜色区分,变元的分类值与变元颜色一致,以便于区分,所有分类值均以其所属变元为圆心进行 Radial Tree 排列,以有效节约视图空间,从而为进一步分析提供了视图基础。验证了 LAMPRT 布局方法取得了较好的效果。

##### 4.2.3 LOVEBRT 算法的案例验证

本小节对 TVBRT 可视化方法进行案例验证,重点结合可视化视图对具体数据集进行深入分析讨论,挖掘数据集中的隐含知识,得出该数据集中隐含的定性结论,达到可视化分析的目的,从而证明了 TVBRT 可视化方法的有效性。

TVBRT 可视化方法对数据集中的层次性分析展示效果更为明显,因此本小节进行案例验证时,从同一变元的两个不同分类值分别开始进行层次化时态分析。对取值较多的变元院校和专业进行研究,院校选取国防科大、空军工程大学为例,专业选取计算机科学与技术、军队指挥类、飞行器动力工程为例。

结合 2010—2012 年的招生数据,使用 TVBRT 可视化方法得到可视化效果图,如图 8 所示,图中的颜色与图 7 的含义相同,虚线表示了不同的分类值节点随时间变化的趋势情况。从图 8 的可视化效果中,至少可以得到以下结论:

- (1) 两所院校在 2011 年招生数均有一定程度的下降。
- (2) 空军工程大学的计算机科学与技术专业 2010 年到 2011 年呈现下降趋势,2012 年该专业不再招生。
- (3) 2011 年,空军工程大学飞行器专业招生数量下降,但

2012年又增加到原先水平。

(4)国防科大计算机科学与技术专业招生呈现下降趋势,2010年到2011年下降较大,2011年与2012年维持平稳。

(5)从2011年开始两所院校招收军队指挥类专业,国防科大当年招生数量非常大、近两年维持平稳,空军工程大学当年招生数较少,2012年出现大幅度增长。

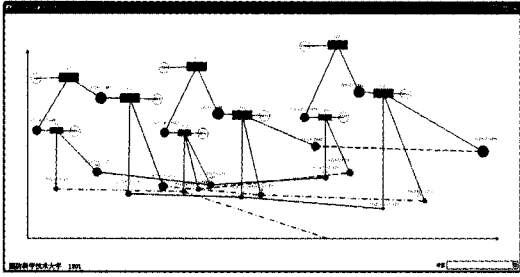


图8 TVBRT可视化方法的案例验证效果图

对上述结论综合分析,可以发现军队院校招生在2011年度进行了相关的调整,有关院校、专业的招生数量出现不同程度的下降,这说明2011年可能存在某些政策方面的调整;某些院校的优势专业在2012年招生数量又恢复到2011年以前的水平,这说明整个招生计划照顾到了相关院校的优势学科,充分说明招生计划的制定具有科学性;某些通用型专业(计算机科学与技术)开始集中在某些院校招生,不具有学科优势的院校不再招生;2011年开始,有关院校开始招收军队指挥类专业的学生,且数量规模非常大,有的院校的军队指挥类招生从2011年到2012年呈现大幅度递增的趋势。

以上完成了使用招生计划数据集作为案例验证TVBRT可视化方法的有效性的过程,可视化效果图能够有效揭示数据集集中的隐含知识。

## 5 TVBRT可视化方法分析

从支持的分析任务、数据特点和视图特点3个方面对本文提出的TVBRT可视化方法进行分析。

(1)支持的分析任务方面,重点进行多变量时态数据的层次性分析,以便灵活地按照不同的变元顺序挖掘数据集中的隐含知识;但变元之间、各分类值之间的度量值的相互关系显示不够明显。

(2)支持的数据特点方面,不同变元或分类值的度量值在同一坐标范围内进行排列,以便明确展现出变元和分类值相互之间的大小关系,这对时态变化趋势显示效果更好;但由于使用多个Radial Tree展示数据集的时态特性,可能造成程序开销较大,因此其适用于时间段不太多的情况。

(3)视图特点方面,视图调整非常灵活,通过3.4节提出的人机交互策略,能够将最关注的对象调整到视图中心,次关注的对象调整在屏幕较为明显的位置;但绘制Radial Tree时,LAMPRT算法需要在具体实现过程中根据可视化效果,指定第一层圆的半径 $R$ 、第二层节点允许存在的角度范围 $\Omega$ 以及第二层圆弧半径的修正值 $r_i$ ;对不同数据集而言,上述参数往往不同,从而引入了一定的主观因素。

**结束语** 本文针对现有具有度量属性的多变量时态数据可视化方法的局限,创新性地经典层次数据可视化方法Radial Tree应用于具有度量属性的多变量时态数据可视化领域,通过提出集成度量属性的Radial Tree布局算法(LAM-

PRT)、基于Radial Tree的具有度量属性的多变量时态数据可视化布局算法(LOVEBRT),并结合相应的人机交互策略,形成了基于Radial Tree的具有度量属性的多变量时态数据可视化方法TVBRT。案例研究表明,TVBRT可视化方法能够综合展现多个变元时态特性,不必进行复杂数据转换,能够开展灵活的分析。

对于TVBRT可视化方法,将来还需要在如何更高效渲染视图(尤其当时间节点进一步增多时,如何表现时态特性)、提高算法效率、进一步完善人机交互等方面进行研究。

## 参考文献

- [1] Guo Dian-sheng, Chen Jin, MacEachren A M, et al. A Visualization System for Space-Time and Multivariate Patterns (VISTAMP)[J]. IEEE Transactions on Visualization and Computer Graphics, 2006, 12(6): 1461-1474
- [2] Guo Dian-sheng, Liao Ke, Morgan M. Visualizing patterns in a global terrorism incident database[J]. Environment and Planning B: Planning and Design, 2007, 34: 767-784
- [3] Akiba H, Ma K-L. An Interactive Interface for Visualizing Time-Varying Multivariate Volume Data[C]//APVIS. 2007
- [4] Akiba H, Ma K-L. A Tri-Space Visualization Interface for Analyzing Time-Varying Multivariate Volume Data[C]//IEEE-VGTC Symposium on Visualization. 2007
- [5] Johansson J, Ljung P, Cooper M. Depth Cues and Density in Temporal Parallel Coordinates[C]//Proceedings of Eurographics/IEEE VGTC Symposium on Visualization. Norrköping, Sweden, 2007, 35-42
- [6] Johansson J. Efficient Information Visualization of Multivariate and Time-Varying Data[D]. Institutionen Teknik Naturvetenskap, 2008
- [7] Lee T-Y, Shen Han-wei. Visualization and Exploration of Temporal Trend Relationships in Multivariate Time-Varying Data[J]. IEEE Transactions on Visualization and Computer Graphics, 2009, 15(6)
- [8] Chen Cheng-kai, Wang Cha-li, Ma K-L, et al. Static Correlation Visualization for Large Time-Varying Volume Data[C]//IEEE Pacific Visualization Symposium 2011. Hong Kong, China, 2011
- [9] Shi Cong-lei, Cui Wei-wei, Liu Shi-xia, et al. RankExplorer: Visualization of Ranking Changes in Large Time Series Data[C]//IEEE Transactions on Visualization and Computer Graphics (InfoVis 2012). 2012
- [10] Greenacre M. Correspondence analysis in practice[M]. Chapman & Hall, 2007
- [11] 孙扬, 唐九阳, 汤大权, 等. 改进的多变元数据可视化方法[J]. 软件学报, 2010, 21(6): 1462-1472
- [12] 威信, 明春英, 王宏安, 等. 基于Venn图的层次信息可视化[J]. 计算机学报, 2007(9): 1632-1637
- [13] 肖卫东, 孙扬, 赵翔, 等. 层次信息可视化技术研究综述[J]. 小型微型计算机系统, 2011(1): 137-146
- [14] Burch M, Weishopf D. Visualizing Dynamic Quantitative Data in Hierarchies-TimeEdgeTrees: Attaching Dynamic Weights to Tree Edges[C]//IVAPP. 2011
- [15] Pavlo A, Homan C, Schull J. A parent-centered radial layout algorithm for interactive graph visualization and animation[OL]. <http://arxiv.org/pdf/cs.HC/0606007>, 2006
- [16] Mansmann S, Scholl M H. Exploring OLAP Aggregates with Hierarchical Visualization Techniques[C]//SAC'07. Seoul, Korea, March, 2007: 11-15

(下转第17页)

测试结果表明,传统调用方法对函数性能的影响更多地受 *mem\_size* 的影响,且较容易出现申请失败的情况,这主要是由于当前运行状态下 *mem\_size* 过大。通过测试可知,如直接使用传统的调用方法,将有 37.25% 的函数无法成功申请到 LDM 空间。同时,传统方法也容易出现不稳定性,如 *sin* 函数基本保持一致;*cot* 函数则通过更多的申请次数申请到了更大的 LDM 空间,但函数性能并没有获得最大的提升;*cos* 函数由于申请次数增大,数据传输时间和申请消耗时间的总和大于 LDM 空间给函数带来的性能提升,最终导致函数性能反而有所下降。

**结束语** 基础函数库在数值计算中发挥着巨大的作用,为大型数值应用提供有效的保障,其性能的高低往往直接或间接地影响着大型应用的实际运行效率。为了提高对基础函数库高性能的要求,本文深入了解体系结构特点,针对体系结构对函数性能的限制,实现了两种有效的访存优化方法,取得了较显著的成效,并将之实际应用于异构从核系统中。

目前,本文主要利用静态程序分析及大量的实验测试实现对核心代码段(计算密集基本块)和访存指令调度位置的判断。而对于高性能的追求是无止境的,同时对于进一步提升函数性能的技术方法也还有很多。但沿着本文的研究方法和思路,下一步的研究重点将主要集中在这两个方面,以实现更精确有效的分析方法,使访存调度工作更精细化,进一步提高函数性能。另外,结合系统实际运行情况,从理论角度实现对 *mem\_size* 和 *N* 关系的分析验证,也是一个新的研究方向,它将为异构众核结构下从核函数库的优化方法提供理论保证。

## 参 考 文 献

- [1] Zhou Hui-yang, Conte T M. Performance modeling of memory latency hiding techniques[R]. Technical report, ECE Department, N. C. State University, January 2003
- [2] Lebeck A R, Koppanalil J, Li T, et al. A large, fast instruction window for tolerating cache misses[C]//Proceedings of the 29th International Symposium on Computer Architecture(ISCA'02). Anchorage, Alaska, USA, IEEE Computer Society, 2002; 59-70
- [3] Wang P H, Wang H, Collins J D, et al. Memory latency-tolerance approaches for itanium processors; out-of-order execution- vs. speculative precomputation[C]//Proceedings of the 8th International Symposium on High Performance Computer Architecture(HPCA'02). Boston, Massachusetts, USA; IEEE Computer Society, 2002; 187-196
- [4] Beyls K, D'Hollander E. Compiler generated multithreading to alleviate memory latency[J]. Journal of Universal Computer Science, 2000, 6(10): 968-993
- [5] 贺红, 朱大铭, 马绍汉. 用神经网络求解时间依赖网络最短路径问题的新算法[J]. 复旦学报: 自然科学版, 2004, 43(5): 714-716
- [6] Raman E, Hundt R, Mannarswamy S. Structure layout optimization for multithreaded programs[C]//Proceedings of the International Symposium on Code Generation and Optimization (CGO'07). San Jose; IEEE Computer Society, 2007; 271-282
- [7] Lattner C, Adve V. Auto-matic pool allocation; improving performance by controlling data structure layout in the heap[C]//Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'05). Chicago, IL, USA; ACM Press, 2005; 129-142
- [8] 黄安文, 高军, 张民选. 多核处理器非一致 Cache 体系结构延迟优化技术研究综述[J]. 计算机研究与发展, 2012, 49(S1): 118-124
- [9] 李浩, 谢伦国. 片上多处理器末级 Cache 优化技术研究综述[J]. 计算机研究与发展, 2012, 49(Suppl1): 172-179
- [10] 余磊, 刘志勇, 宋凤龙. LU 分解在众核结构仿真器上的指令级调度研究[J]. 系统仿真学报, 2011, 23(12): 2603-2610
- [11] Allen R, Kennedy K. Optimizing Compilers for Modern Architectures, A Dependence-Based Approach[M]//Elsevier Science, 2004; 47-374
- [12] Zhao Jie, Zhao Rong-cai, Han Lin. A Nonlinear Array Subscripts Dependence Test[C]//Proceedings of the 2012 IEEE 14th International Conference on High Performance Computing and Communications(HPC'12). Liverpool, IEEE Computer Society, 2012; 764-771
- [13] Rau B R, Fisher J A. Instruction level parallel-processing; history, overview and perspective[J]. The Journal of Supercomputing, 1993, 7(1): 950
- [14] Garey M R, Johnson D S. Computers and Intractability; A Guide to the Theory of NP-Completeness[M]. Freeman W H. Co, San Francisco, 1979
- [15] using treemaps[J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(6): 1286-1293
- [16] Burch M, Beck F, Diehl S. Timeline Trees; Visualizing sequences of transactions in information hierarchies[C] // International Working Conference on Advanced Visual Interfaces (AVI'08). 2008; 75-82
- [17] Yee Ka-ping, Fisher D, Dhamija R, et al. Animated exploration of dynamic graphs with radial layout[C]//Proceedings of the IEEE Symposium on Information Visualization. 2001; 43-50
- [18] Book G, Keshary N. Radial Tree graph drawing algorithm for representing large hierarchies[D]. University of Connecticut, December 2001
- [19] Sheth N, Cai Q. Visualizing mesh dataset using radial tree layout [R/OL]. Spring 2003 Information Visualization Class Project, Indiana University, 2003
- [20] Douma M, Ligierko G, Ancuta O, et al. SpicyNodes; Radial Layout Authoring for the General Public[J]. IEEE Transactions on Visualization and Computer Graphics, 2009, 15(6): 1089-1096
- [21] Card S K, Suh B, Pendleton B A, et al. Time Tree; Exploring time changing hierarchies[C]//IEEE Symposium on Visual Analytics Science and Technology (VAST'06). 2006; 3-10
- [22] Tu Y, Shen H-W. Visualizing changes of hierarchy information using treemaps[J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(6): 1286-1293
- [23] Burch M, Beck F, Diehl S. Timeline Trees; Visualizing sequences of transactions in information hierarchies[C] // International Working Conference on Advanced Visual Interfaces (AVI'08). 2008; 75-82
- [24] Burch M, Diehl S. TimeRadarTrees; Visualizing dynamic compound digraphs[J]. Computer Graphics Forum, 2008, 27(3): 823-830
- [25] <http://prefuse.org>
- [26] Heer J, Card S K, Landay J A. Prefuse; a toolkit for interactive information visualization[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York; ACM Press, 2005; 421-430
- [27] 孙宁伟, 刘海峰, 赵瑜, 等. TVBPS; 一种基于 Parallel Sets 的具有度量属性的多变元时态数据可视化方法[J]. 计算机应用研究, 2014, 31(5): 1591-1596

(上接第 11 页)

- [17] Yee Ka-ping, Fisher D, Dhamija R, et al. Animated exploration of dynamic graphs with radial layout[C]//Proceedings of the IEEE Symposium on Information Visualization. 2001; 43-50
- [18] Book G, Keshary N. Radial Tree graph drawing algorithm for representing large hierarchies[D]. University of Connecticut, December 2001
- [19] Sheth N, Cai Q. Visualizing mesh dataset using radial tree layout [R/OL]. Spring 2003 Information Visualization Class Project, Indiana University, 2003
- [20] Douma M, Ligierko G, Ancuta O, et al. SpicyNodes; Radial Layout Authoring for the General Public[J]. IEEE Transactions on Visualization and Computer Graphics, 2009, 15(6): 1089-1096
- [21] Card S K, Suh B, Pendleton B A, et al. Time Tree; Exploring time changing hierarchies[C]//IEEE Symposium on Visual Analytics Science and Technology (VAST'06). 2006; 3-10
- [22] Tu Y, Shen H-W. Visualizing changes of hierarchy information