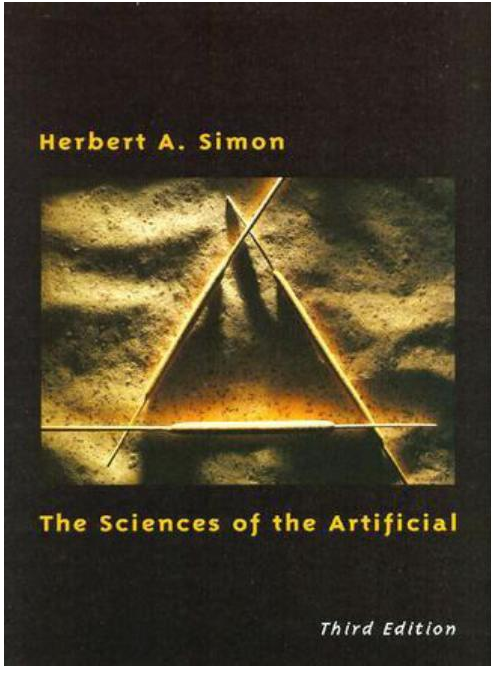
第8章 层次和网络数据可视化

1 层次数据

为什么要层次结构？

- 层次结构是复杂系统中有效的管理组织形式，以及稳定、可持续的发展机制
- "The Architecture of Complexity: Hierarchic Systems"
  - 社会系统
  - 生物和物理系统
  - 符号系统



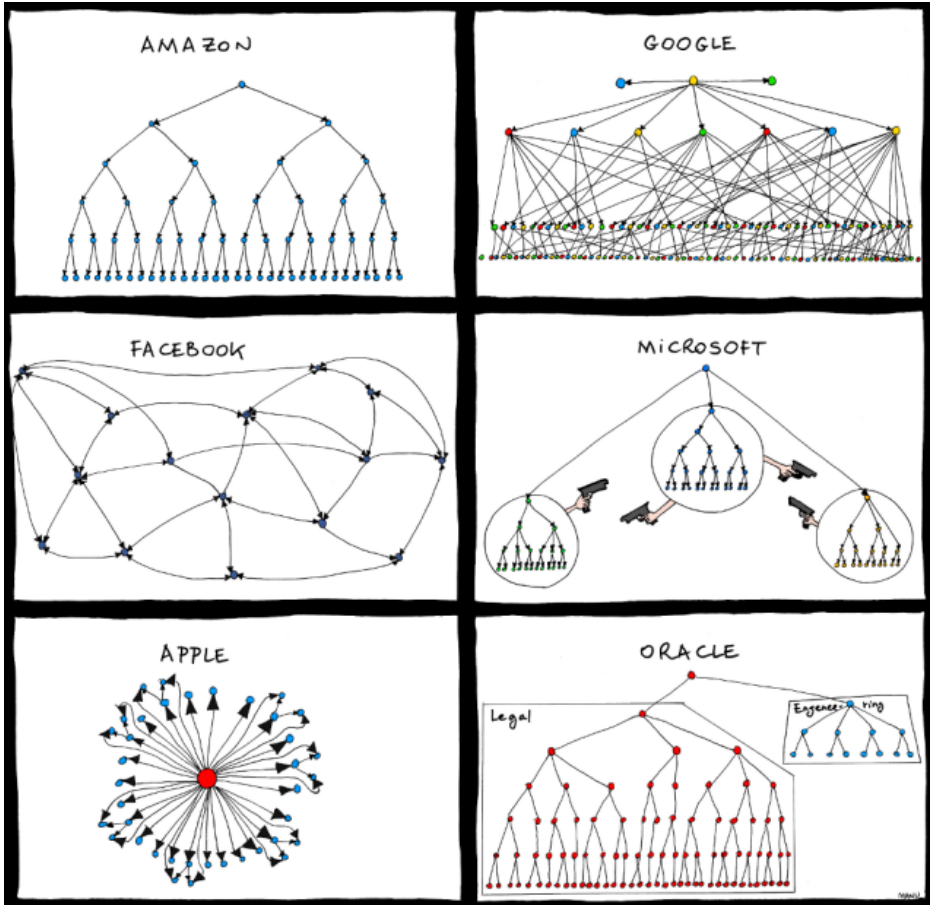
流水线模型

- 层次数据是一种常见的数据类型，着重表达个体之间的层次关系
  - 这种关系主要表现为两类：包含和从属

层次数据

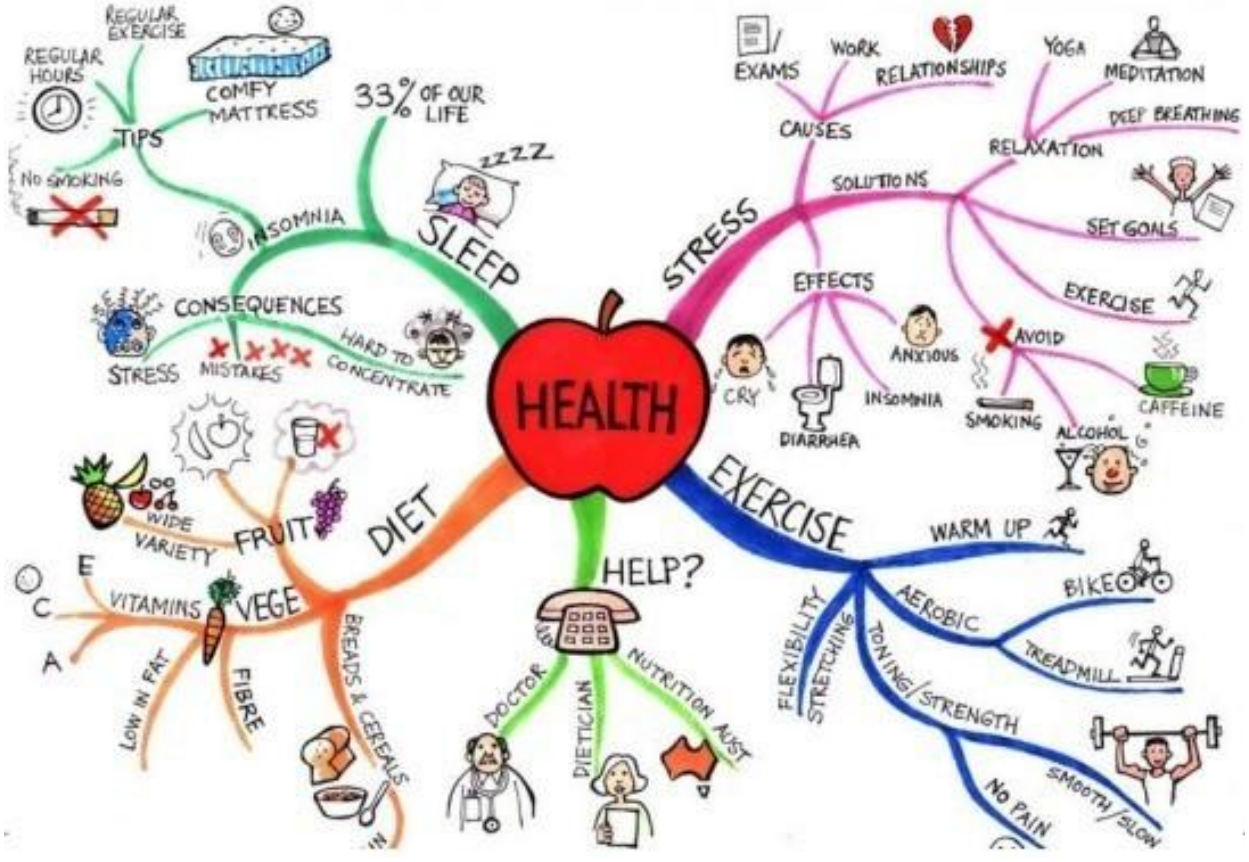
- 层次数据着重表现个体之间的隶属关系
  - 机构的组织结构，物种关系

层次数据

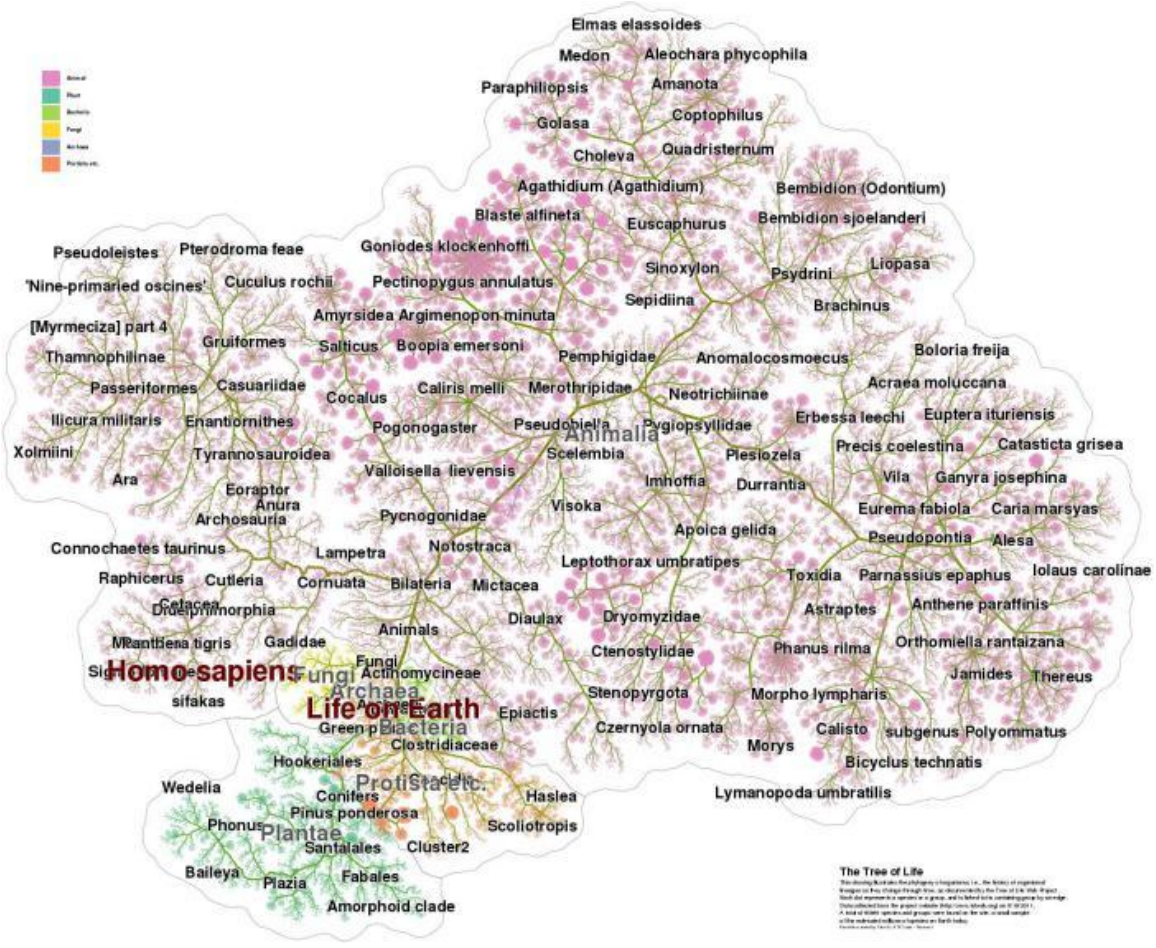

AMAZON
GOOGLE
FACEBOOK
MICROSOFT
APPLE
ORACLE
Legal
Engineering

层次数据

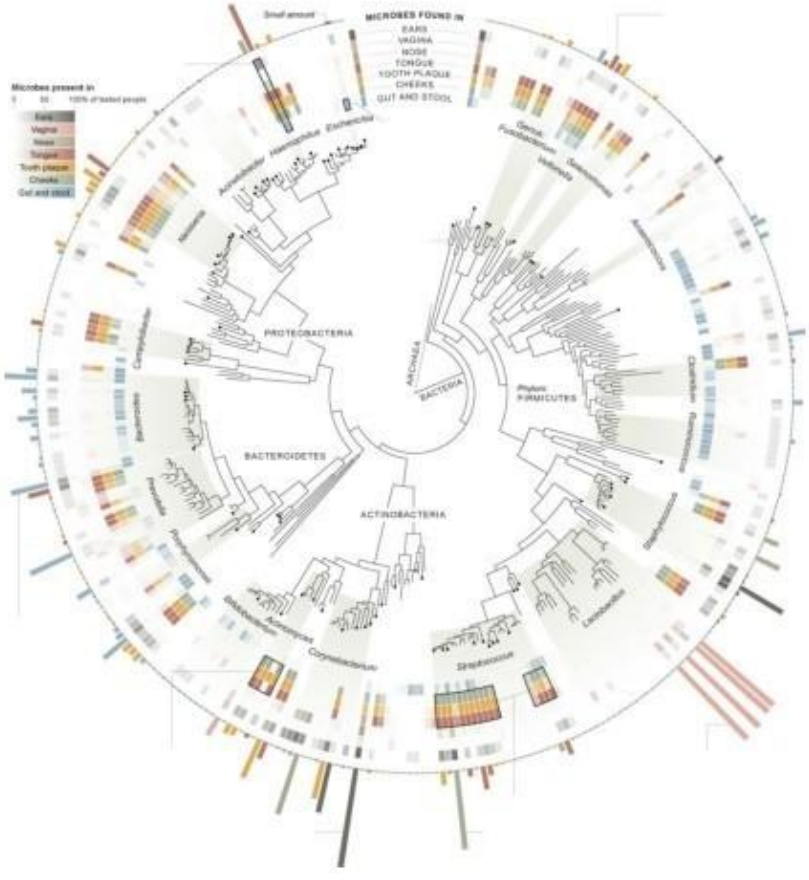
- “我们通过分类来理解事物，层次结构是我们认知行为的基础”
  - 认知信息：文件列表
  - 进行记忆和思维发散时：思维导图
  - 多层级的结构对个体进行分类：图书文献、分类学、物种发展史等学科的核心


HEALTH
SLEEP
33% OF OUR LIFE
ZZZZ
INSOMNIA
TIPS
REGULAR EXERCISE
REGULAR HOURS
COMFY MATTRESS
NO SMOKING
CONSEQUENCES
STRESS
MISTAKES
HARD TO CONCENTRATE
STRESS
CAUSES
EXAMS
WORK
RELATIONSHIPS
SOLUTIONS
RELAXATION
YOGA
MEDITATION
DEEP BREATHING
SET GOALS
EXERCISE
EFFECTS
CRY
DIARRHEA
INSOMNIA
ANXIOUS
AVOID
SMOKING
CAFFEINE
ALCOHOL
EXERCISE
WARM UP
BIKE
AEROBIC
TREADMILL
TONING/STRENGTH
SMOOTH/SLOW
NO PAIN
FLEXIBILITY
STRETCHING
DIET
FRUIT
WIDE VARIETY
VEGE
VITAMINS
C
E
A
LOW IN FAT
FIBRE
BREADS & CEREALS
HELP?
DOCTOR
DIETICIAN
NUTRITION AUST

物种发展

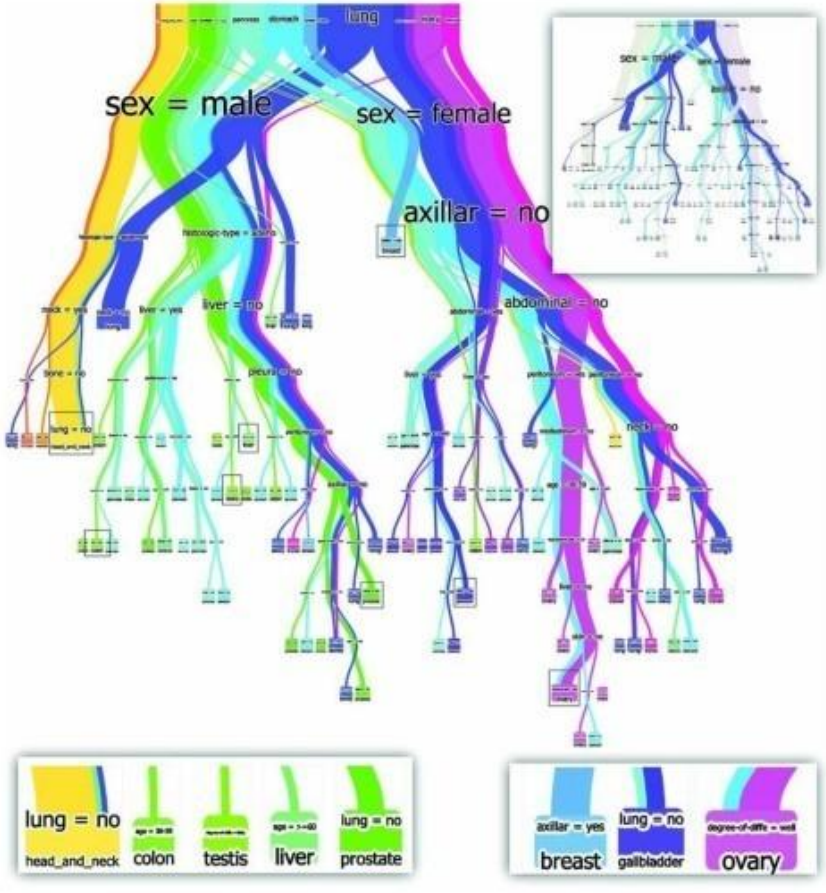

Elmas elassoides
Medon
Aleochara phycophila
Paraphiliopsis
Amanota
Golasa
Coptophilus
Choleva
Quadristernum
Agathidium (Agathidium)
Bembidion (Odontium)
Blaste alfineta
Euscaphurus
Bembidion sjoelanderi
Pseudoleistes
Pterodroma feae
Goniodes klockenhofft
Sinoxylon
Psydrini
Liopasa
'Nine-primaried oscines'
Cuculus rochii
Pectinopygus annulatus
Sepidiina
Brachinus
[Myrmeciza] part 4
Amyrsidea Argimenopon minuta
Thamnophilinae
Gruiformes
Salticus
Boopia emersoni
Pemphigidae
Anomalocosmoecus
Boloria freija
Passeriformes
Casuariidae
Cocalus
Caliris melli
Merothripidae
Neotrichiinae
Acraea moluccana
Ilicura militaris
Enantiornithes
Pogonogaster
Pseudobiella
Pygiopsyllidae
Erbessa leechi
Euptera ituriensis
Animalia
Xolmiini
Tyrannosauroidea
Scelembia
Plesiozela
Precis coelestina
Catasticta grisea
Ara
Valloisella lievensis
Imhoffia
Durrantia
Vila
Ganyra josephina
Eoraptor
Anura
Archosauria
Visoka
Eurema fabiola
Caria marsys
Pycnogonidae
Apolca gelida
Pseudopontia
Alesa
Connochaetes taurinus
Lampetra
Notostraca
Leptothorax umbratipes
Raphicerus
Cutleria
Cornuata
Bilateria
Mictacea
Toxidia
Parnassius epaphus
Iolaus carolinae
Cetacea
Didelphimorphia
Diaulax
Dryomyzidae
Astraptes
Anthene paraffinis
Animals
Maanthiena tigris
Gadidae
Ctenostylidae
Phanus rilma
Orthomiella rantaizana
Fungi
Homo sapiens
Actinomycineae
Archaea
Life on Earth
Stenopyrgota
Morpho lympharis
Jamides
Thereus
sifakas
Epiactis
Green p
Bacteria
Czernyola ornata
Calisto
subgenus Polyommatus
Clostridiaceae
Morys
Bicyclus technatis
Hookeriales
Protista
etc.
Wedelia
Conifers
Haslea
Lymanopoda umbratilis
Pinus ponderosa
Phonus
Plantae
Scoliotropis
Santalales
Cluster2
Baileya
Plazia
Fabales
Amorphoid clade
The Tree of Life

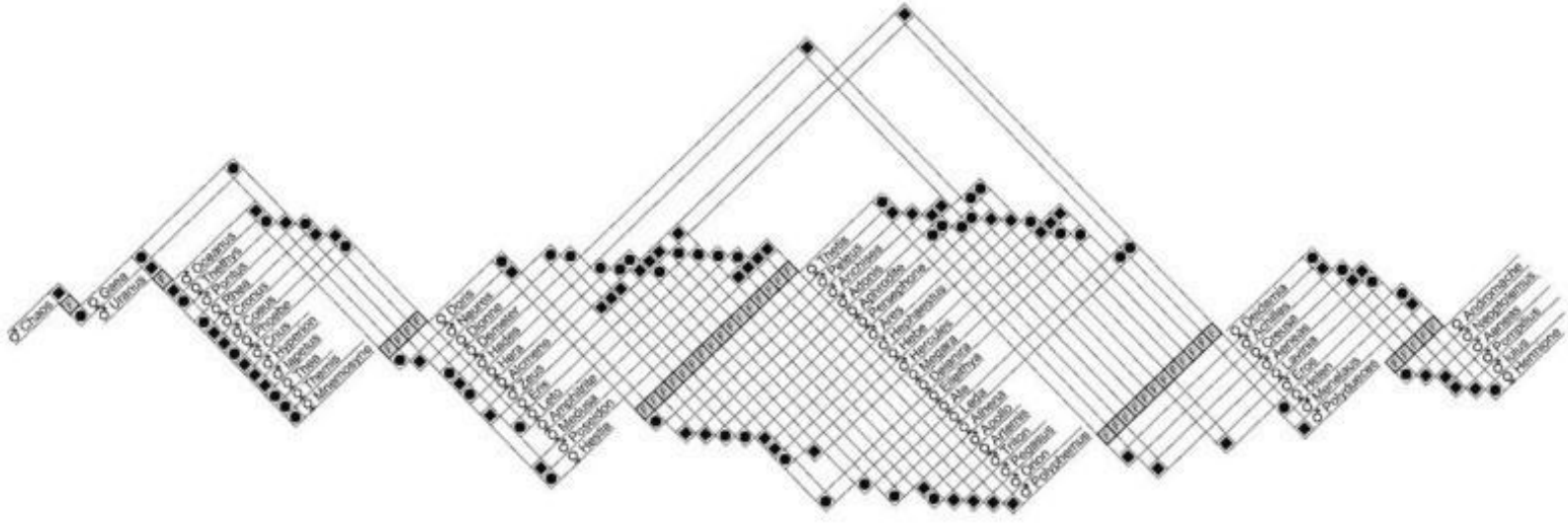
物种发展


MICROBES FOUND IN
EARS
VAGINA
NOSE
TONGUE
TOOTH PLAQUE
CHEEKS
GUT AND STOOL
Microbes present in
Ears
Vagina
Nose
Tongue
Tooth plaque
Cheeks
Gut and stool
ARCHAEA
BACTERIA
PROTEOBACTERIA
BACTEROIDETES
ACTINOBACTERIA
FIRMICUTES
Escherichia
Haemophilus
Acinetobacter
Neisseria
Campylobacter
Bacteroides
Prevotella
Porphyromonas
Capnocytophaga
Actinomyces
Corynebacterium
Streptococcus
Leptotrichia
Staphylococcus
Lactobacillus
Veillonella
Gemella
Granulicatella
Clostridium

层次数据

- 表示逻辑上的承接关系
  - 机器学习中的决策树，每一个节点就是一个问题，不同答案对应不同的分支，连接到下一层的子节点。最底层的叶节点则通常对应最后的决策
  - 家谱


sex = male
sex = female
axillar = no
liver = no
abdominal = no
lung = no
head_and_neck
colon
testis
liver
prostate
axillar = yes
breast
lung = no
gallbladder
ovary


Chaos
Gaea
Uranus
Oceanus
Thethys
Pontus
Pheie
Coeus
Phoebe
Crius
Hyperion
Iapetus
Thein
Themis
Mnemosyne
Dione
Neurite
Dionne
Demeter
Lebes
Hera
Alcmene
Zeus
Leto
Artemis
Nedume
Poseidon
Hestia
Thetis
Peleus
Artemis
Metis
Aphrodite
Persephone
Ares
Hephaestus
Hebe
Hercules
Megara
Deianira
Dexithya
Ate
Leto
Athena
Apollo
Artemis
Hiton
Prometheus
Orion
Polydeuces
Deidamia
Achilles
Ceteus
Aerope
Linus
Eros
Helen
Menelaus
Polydeuces
Andromache
Neoptolemus
Hermes
Pompeius
Iulus
Hermione

树与图

- 层次结构可以被抽象成树型(Tree)结构，它是以分支关系定义的非线性结构
- 树是图(Graph)的一个特例，只存在父节点和子节点之间连接的图
  - 在一棵树里，只有根节点没有父节点，其余节点有且仅有一个与之相连的父节点
  - 没有子节点的节点称为叶节点，同一个层次具有相同父节点的节点称为兄弟节点
  - 每一个节点都可以有若干个子节点
  - 从根节点到某个特定节点之间的连接数量，称为该节点的深度；具有相同深度的节点数，称为该层的广度

- 层级结构中的个体元素
  - 相关的子系统(interrelated subsystems)
  - 相对独立(semi-independent)
- 子系统之间的关系
  - 影响往往是通过子系统之间的功能(输入、输出)
  - 和一个子系统内部实现其功能的具体方式无关
- 可拆分(decomposability)或准可拆分(near decomposability

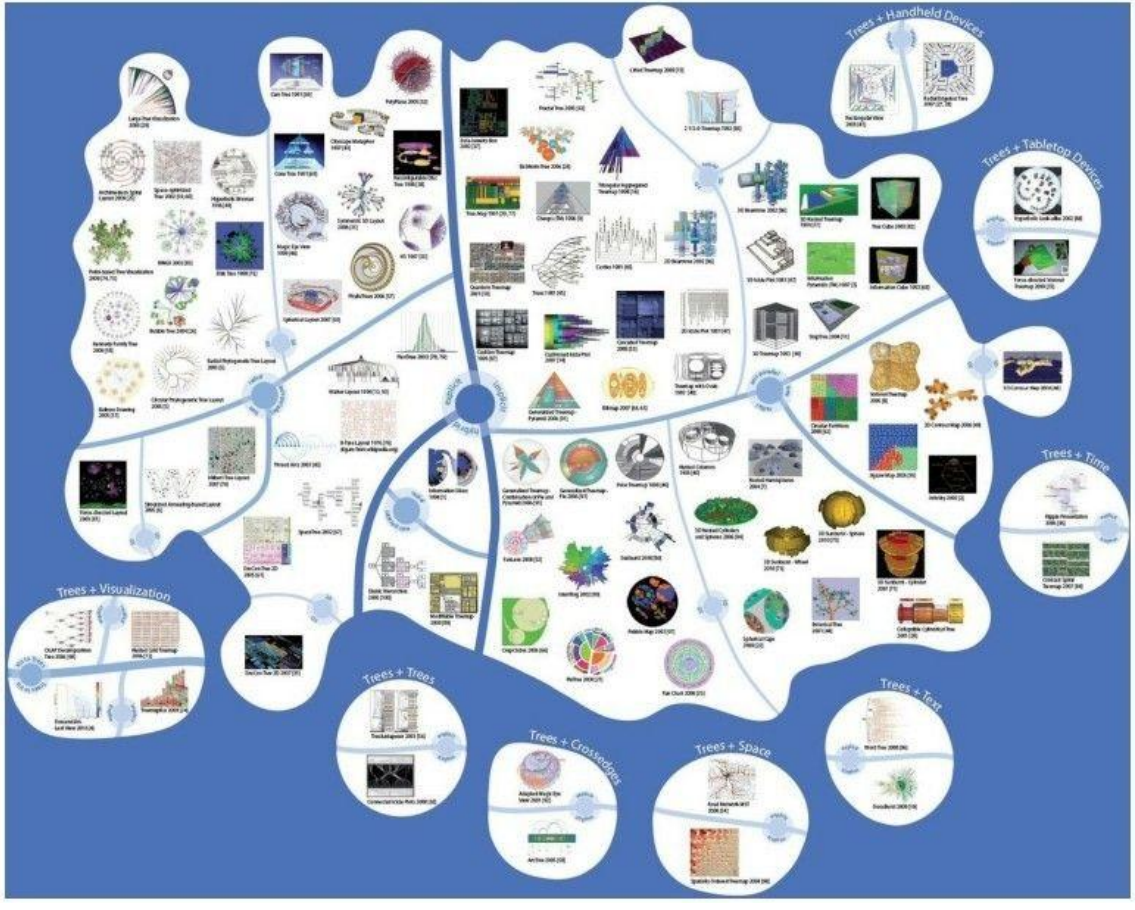
主要的挑战

- 把节点和边信息展示出来
  - 点和边的空间排布组织形式
- 允许用户对层次数据进行交互式分析探索
  - 对层次数据的相关部分进行观察分析

- 展示：合理的利用显示空间
- 交互：和任务相关的功能工具
  - 微观细节和宏观背景
  - 对子系统进行汇总、比较不同的子系统等

层次数据的可视化

- 层次关系(即树型结构)的有效刻画，采用不同的视觉符号来表示不同类型的关系：
  - 节点-链接(Node-link)
    - 将单个个体绘制成一个节点，节点之间的连线表示个体之间的层次关系
  - 空间填充(Space-filling)
    - 用空间中的分块区域表示数据中的个体，并用外层区域对内层区域的包围表示彼此之间的层次关系
  - 混合前两种方法的思路


Trees + Handheld Devices
Trees + Tableton Devices
Trees + Time
Trees + Text
Trees + Space
Trees + Crossedges
Trees + Trees
Trees + Visualization

节点-链接法

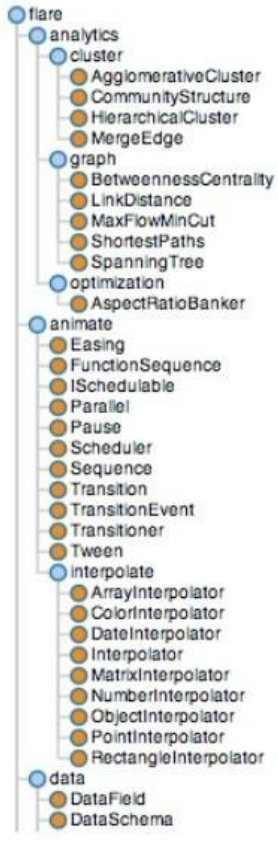
- 节点-链接法的核心问题，如何在屏幕上放置节点，以及如何绘制节点及节点之间的链接关系
  - 节点位置的空间顺序和层次关系一致
  - 减少连线之间的交叉。过多的连线交叉会干扰用户对关系的解读
  - 减少连线的总长度。连线越长越容易造成解读错误
  - 可视化应该有一个合适的长宽比，以便优化空间的利用

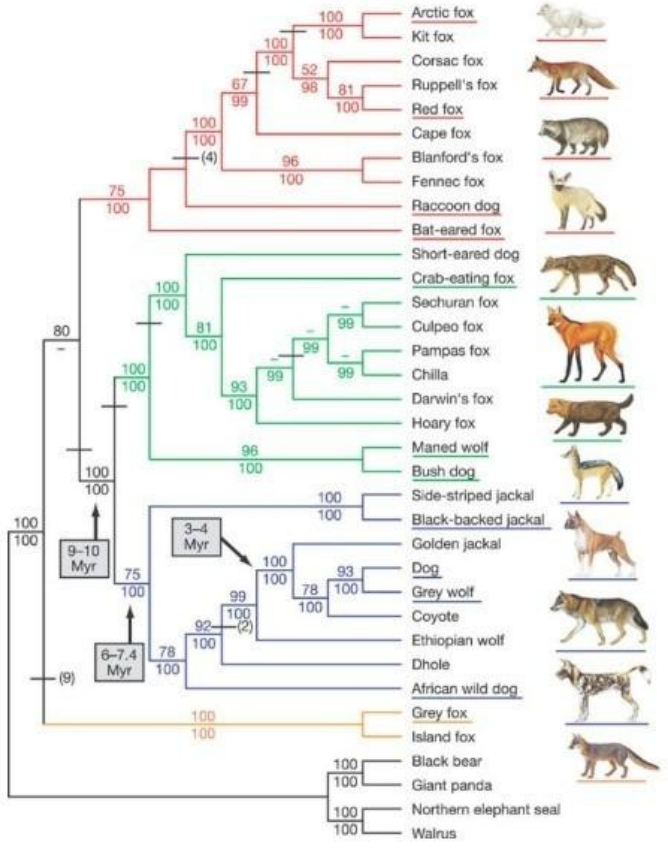
- 可视化要求有时互相冲突
  - 一个好的节点-链接布局算法要满足尽量多的要求，且不同的应用侧重于不同的布局要求

缩进图

- 快速并易于实现
- 可以使用纯文本(或HTML)

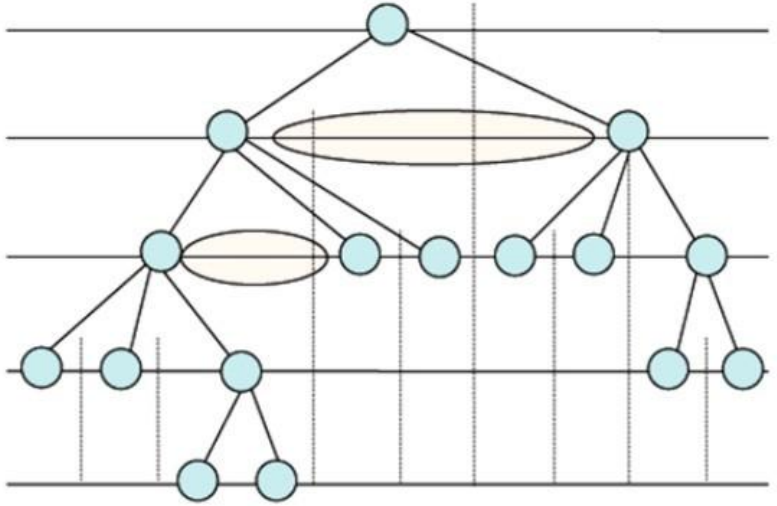
- 浏览大数据时需要很多滚动操作
- 容易失去上下文(context





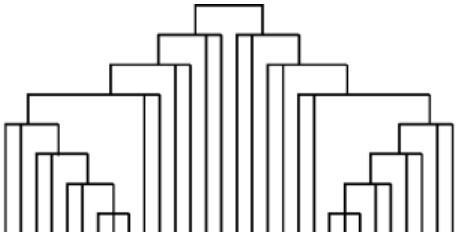
普适的纵横轴布局算法

- (1)根据树的深度将空间沿纵轴平均分成等高的区域，每个区域对应树的一层。相同深度的节点属于同一层
- (2)根据每一层节点的数量，将对应的区域沿横轴平均分成等宽的区域
- (3)将节点布置在每个区域的中心
- (4)在节点和它的父节点之间连线

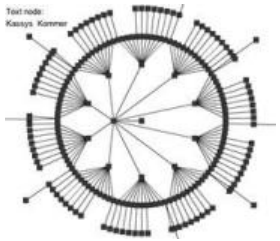


节点与链接的布局

- 正交布局
  - 缩进图(indent)
  - 聚类树(dendrogram)
  - 冰柱图(icicle)
- 径向布局(辐射型)
  - 径向布局图
  - 双曲树
- 自由布局



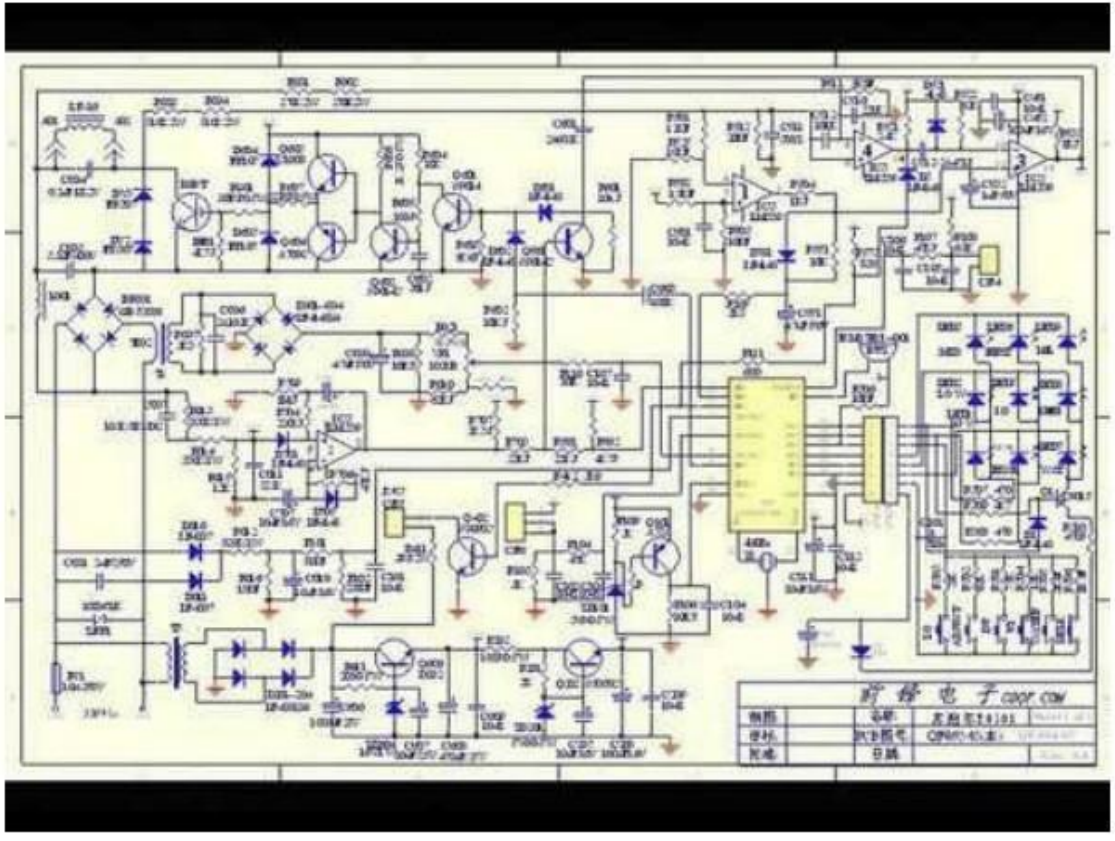
dendrogram



径向布局

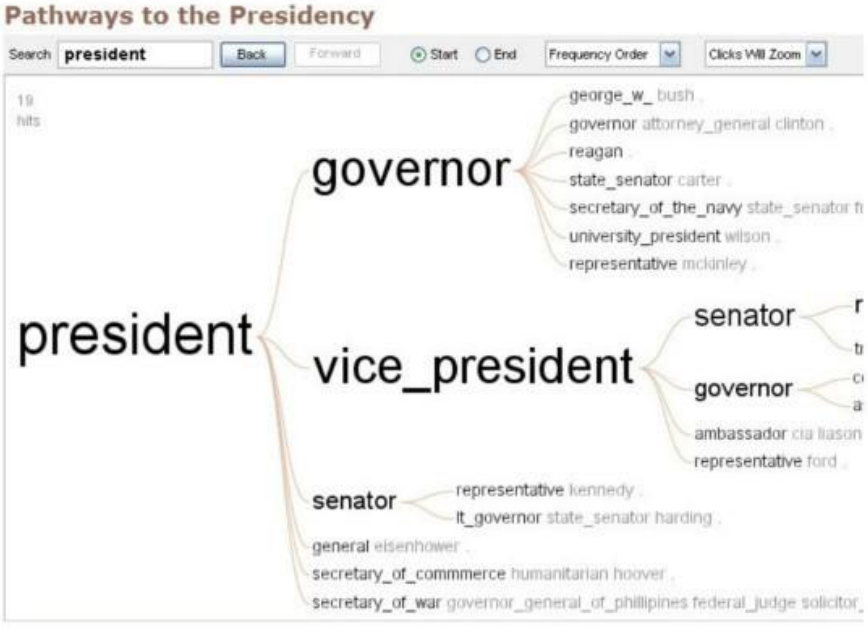
正交布局

- 节点在放置的时候都按照水平或垂直对齐
- 方向与坐标轴一致的，布局规则
- 与视觉识别习惯吻合，非常直观
- 缺点
  - 对于大型的层次结构，特别是广度比较大的层次结构，这样的布局会导致不合理的长宽比

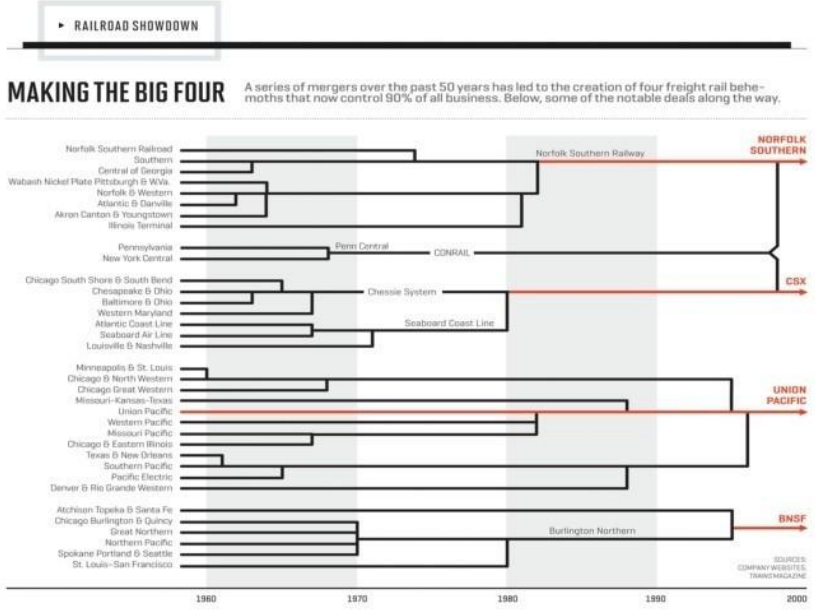


应用

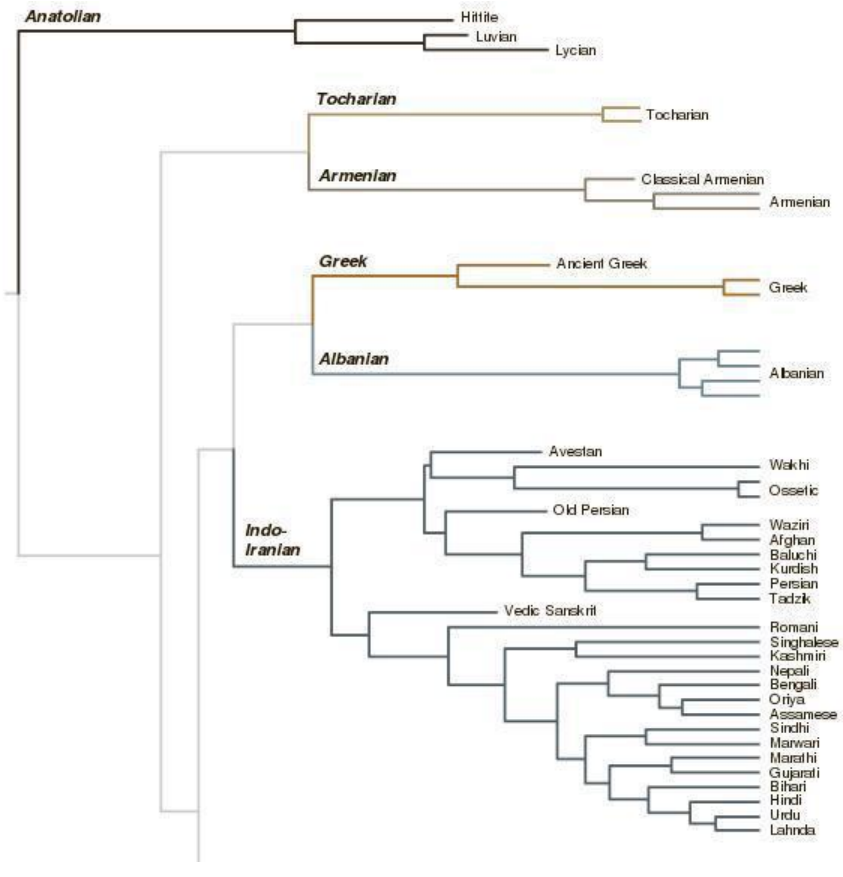
- 单词树可视化设计



- 美国铁路的兼并

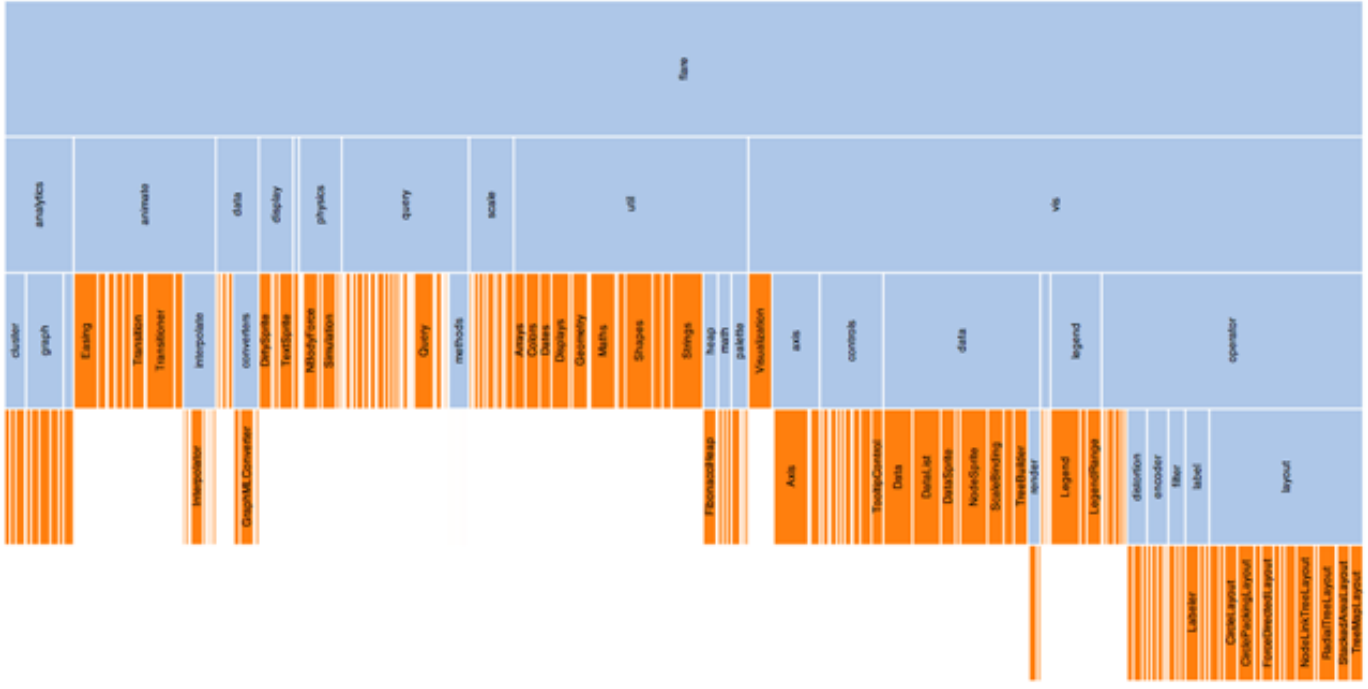


印欧语系


Anatolian
Hittite
Luvian
Lycian
Tocharian
Tocharian
Armenian
Classical Armenian
Armenian
Greek
Ancient Greek
Greek
Albanian
Albanian
Indo-
Iranian
Avestan
Wakhi
Ossetic
Old Persian
Waziri
Afghan
Baluchi
Kurdish
Persian
Tadzik
Vedic Sanskrit
Romani
Singhalese
Kashmiri
Nepali
Bengali
Oriya
Assamese
Sindhi
Marwari
Marathi
Gujarati
Bihari
Hindi
Urdu
Lahnda

冰柱图

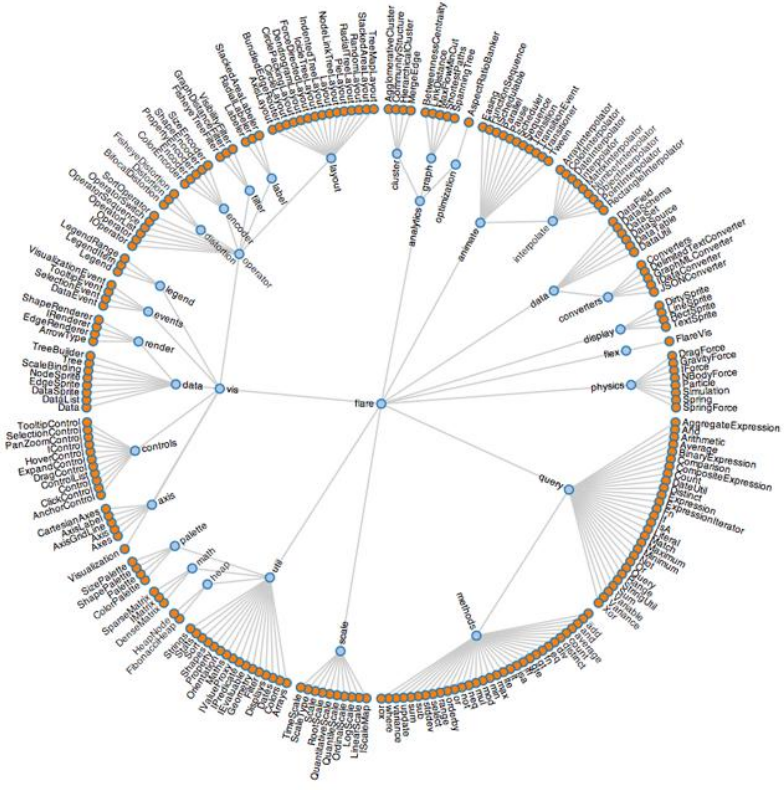
- 常用于聚类分析，展现层次聚类结果



径向布局

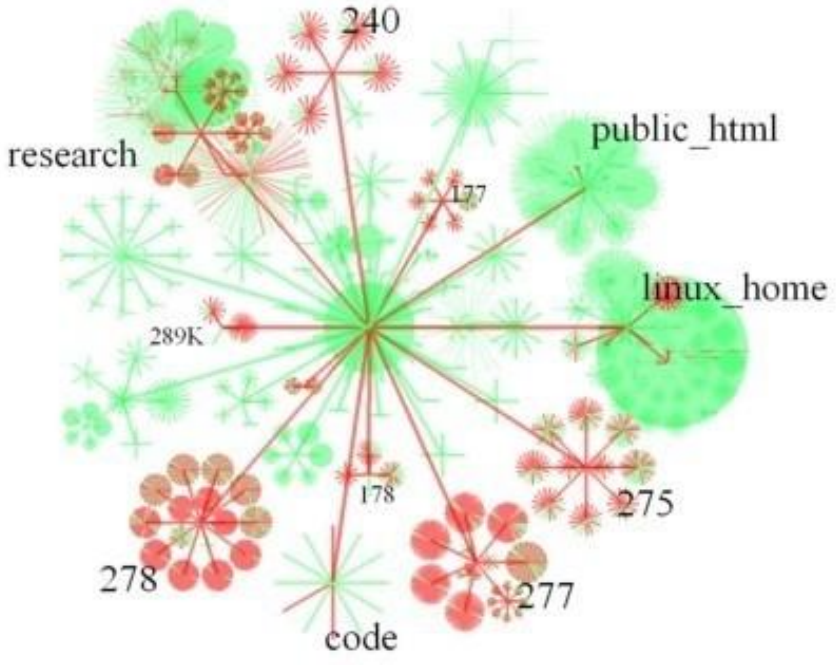
- 更加合理地利用空间
- 根节点位于圆心，不同层次的节点被放置在半径不同的同心圆上
- 节点到圆心的距离对应于它的深度
- 满足树结构的节点数量随层次增加

Flare软件包的目录结构



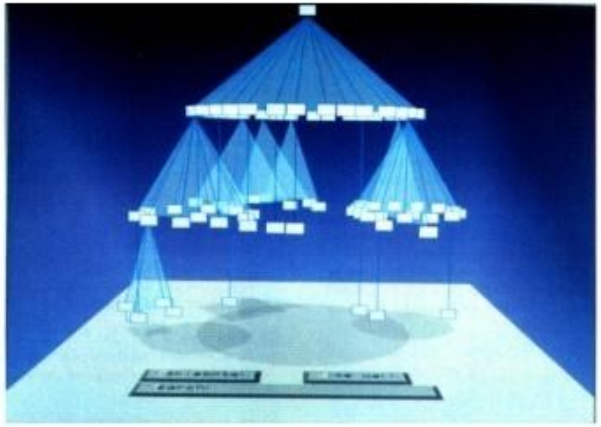
径向树扩展

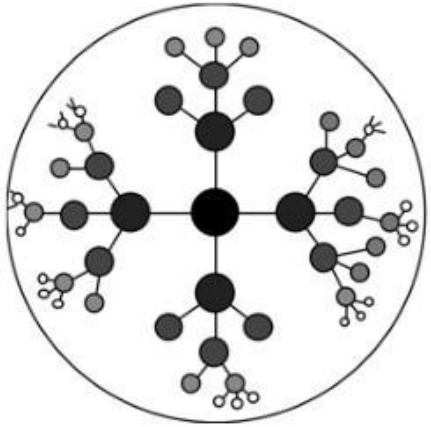
- 每一棵子树递归地采用径向布局，形成环状结构。使得子树的结构更加直观



圆锥树(Cone Tree)

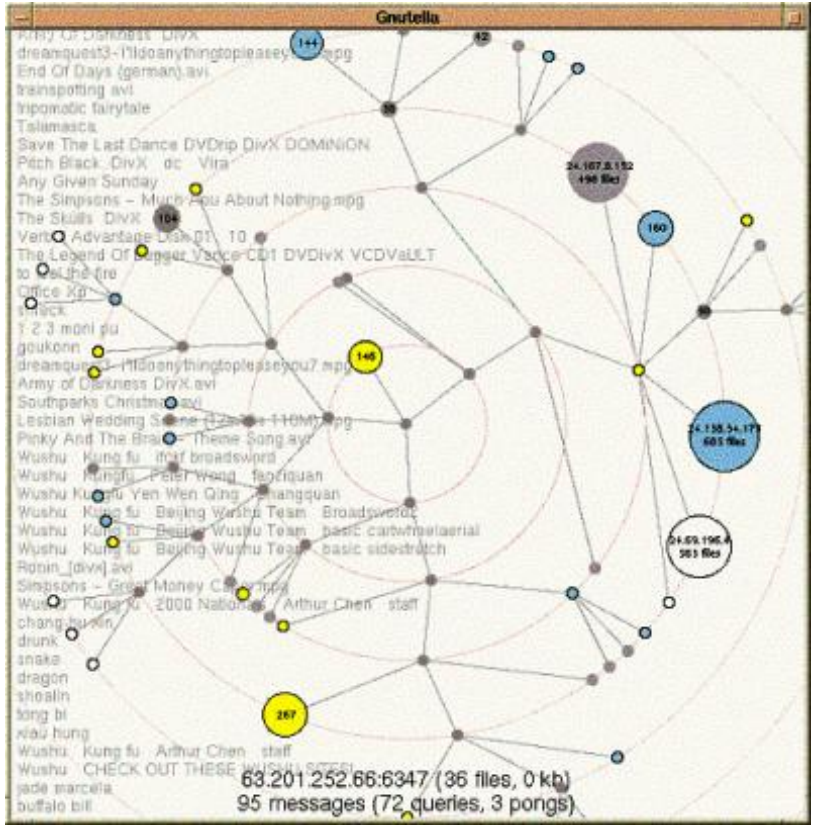
- 一种在三维空间可视化层次数据的技术
- 结合了径向布局和正交布局两种思想
  - 在每一层上，属于同一个父节点的子节点沿着以父节点为圆心的圆呈放射状排列
  - 不同层次被放置在空间中不同的高度，因此形成了以父节点为顶点，子节点放置在底部的圆锥





径向布局

- Radial



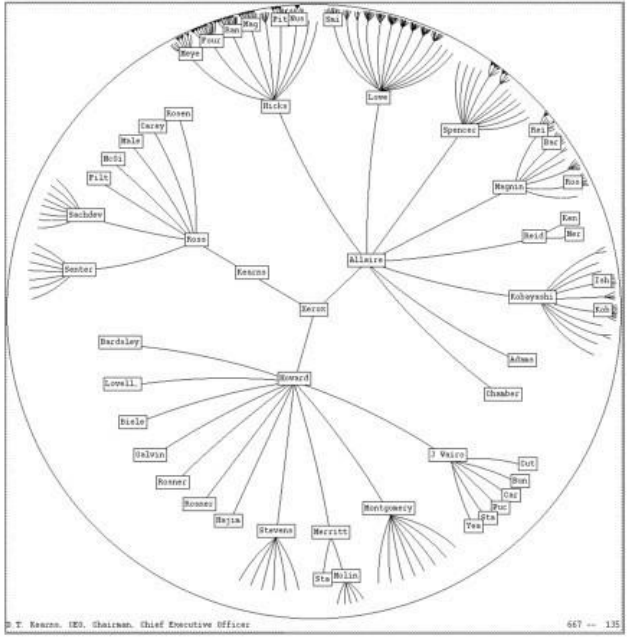
http://flare.prefuse.org/demo

三维树(Cone-Tree)

- 三维空间，结合二维投影
- 优点
  - 三维空间来扩展可用显示空间
  - 三维动画来降低认知成本
- 缺点
  - 难以对付很大的树
  - 三维交互还是一个挑战

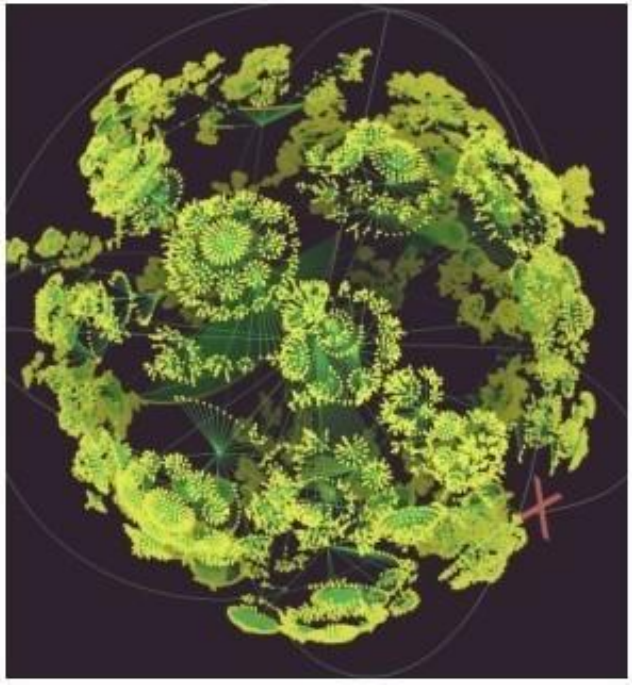
双曲树(Hyperbolic Tree)

- 利用对二维空间的非线性映射来有效地利用空间
- 拓扑结构和常规的径向树方法一致，布局空间是双曲空间(Hyperbolic Space)



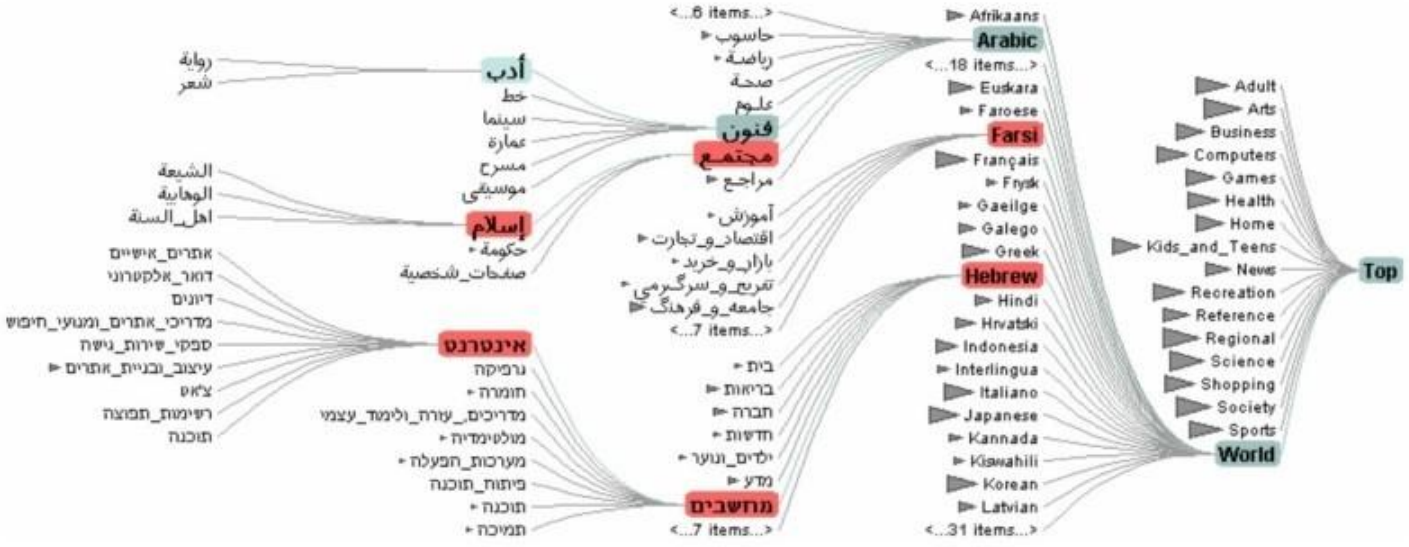
双曲树(Hyperbolic Tree)

- 在三维空间中同样可以采用双曲树方法



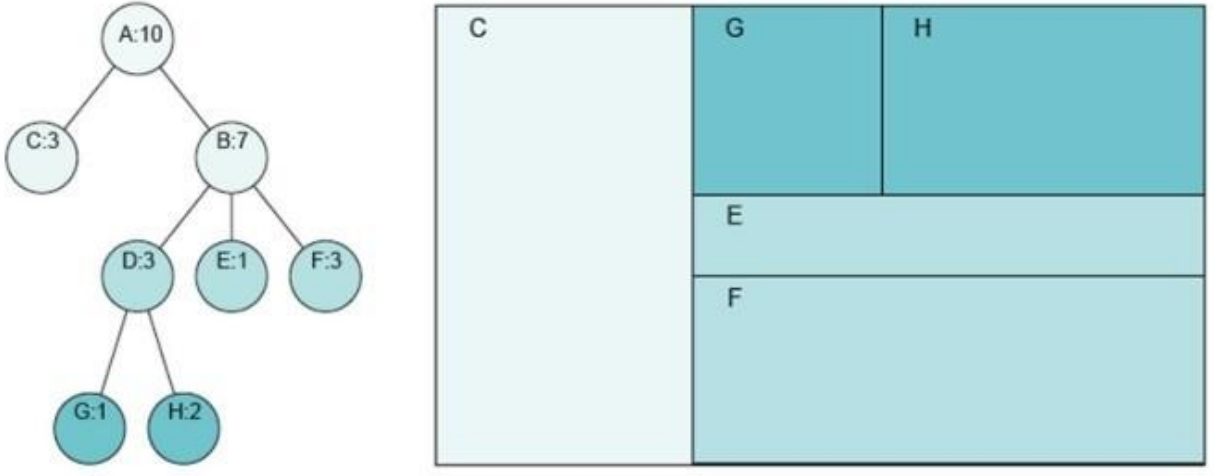
“焦点+上下文”(Focus+Context)技术

- 大尺度的层次结构，由于屏幕尺寸的限制，无法避免节点的重叠
  - 有效地利用空间，重点呈现用户关心的重要数据，同时简要地表现上下文信息



空间填充法

- Johnson和Schneiderman在1991年发明的树图(Treemap)
- 从空间填充的角度实现层次数据的可视化
- 树图法采用矩形表示层次结构里的节点，父子节点之间的层次关系用矩形之间的相互嵌套隐喻来表达



```
Treemap 函数
    设置初始宽度为可视化空间的宽度
    设置初始高度为可视化空间的高度
    设置初始节点为树的根节点
    设置初始位置为起点，如 (0, 0)
    设置初始方向为水平方向或竖直方向
    调用 Divide 函数
Treemap 函数结束

Divide 函数（节点，方向，位置，宽度，高度）
    IF 该节点是叶节点
        按该节点被赋的长宽和位置画一个长方形
        返回

    FOR 该节点的每一个子节点 i
    计算子节点 i 在所有子节点中的比重
    IF 该节点是水平方向
        子节点的宽度：= 该节点宽度 * 子节点的比重
        子节点的方向为竖直方向
        递归进行 Divide 函数
        子节点位置：= 该节点的位置 + 子节点的宽度
    IF 该节点是竖直方向
        子节点的高度：= 该节点高度 * 子节点的比重
        子节点的方向为水平方向
        递归进行 Divide 函数
        子节点位置：= 该节点的位置 + 子节点的高度
    ENDFOR
Divide 函数结束
```

树图可视化

树图可视化的一个核心问题是在递归地分割空间时采用分割方法分割得到的矩形树图的视觉质量通常通过矩形的长宽比来度量

Johnson和Schneiderman发明的交替纵横切分法(Slice-and-Dice).在整个递归过程中，空间依次交替着被横切或者纵切。当某个层次的一个节点有大量子节点时，严格遵循交替纵横切分法会导致非常细的条状空间，影响对数据的视觉理解

正等分法(Squarified)可有效地改进视觉体验和可视化效果 [Bruls1999]。算法力图使分割出来的区域尽量接近方形，同时采用一种贪心策略来达到更高的效率和次优的结果

。。。

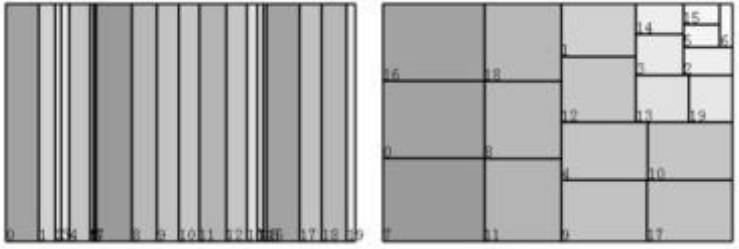
De Berg、Speckmann和Van Der Weele[Berg2011]证明了矩形树图的纵横比最小化是强NPC问题

当树图的输入数据发生变化时，则需要通过另一个质量标准对生成的树图进行度量，即树图的稳定性——树图随着数据的变化会发生多大的变化

Shneiderman和Wattenberg[Shnei2001]提出了第一种需要考虑稳定性的树图——有序树图

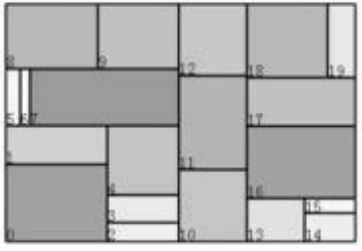
Shneiderman和Wattenberg[Shnei2001]提出了Pivot-by-(Middle,Size and Split-Size)算法。Bederson、Shneiderman和Wattenberg[Beder2002]提出了Strip算法，沿固定方向一次一条带地划分

常见的6种空间划分方法生成的树图效果

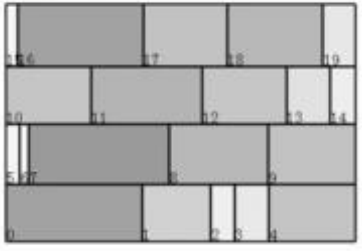


Slice-and-dice

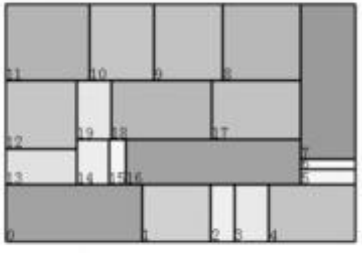
Squarified



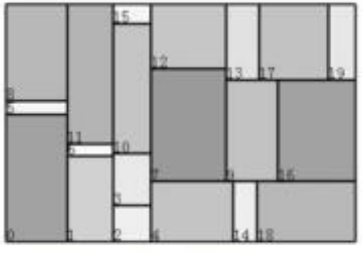
Ordered



Strip

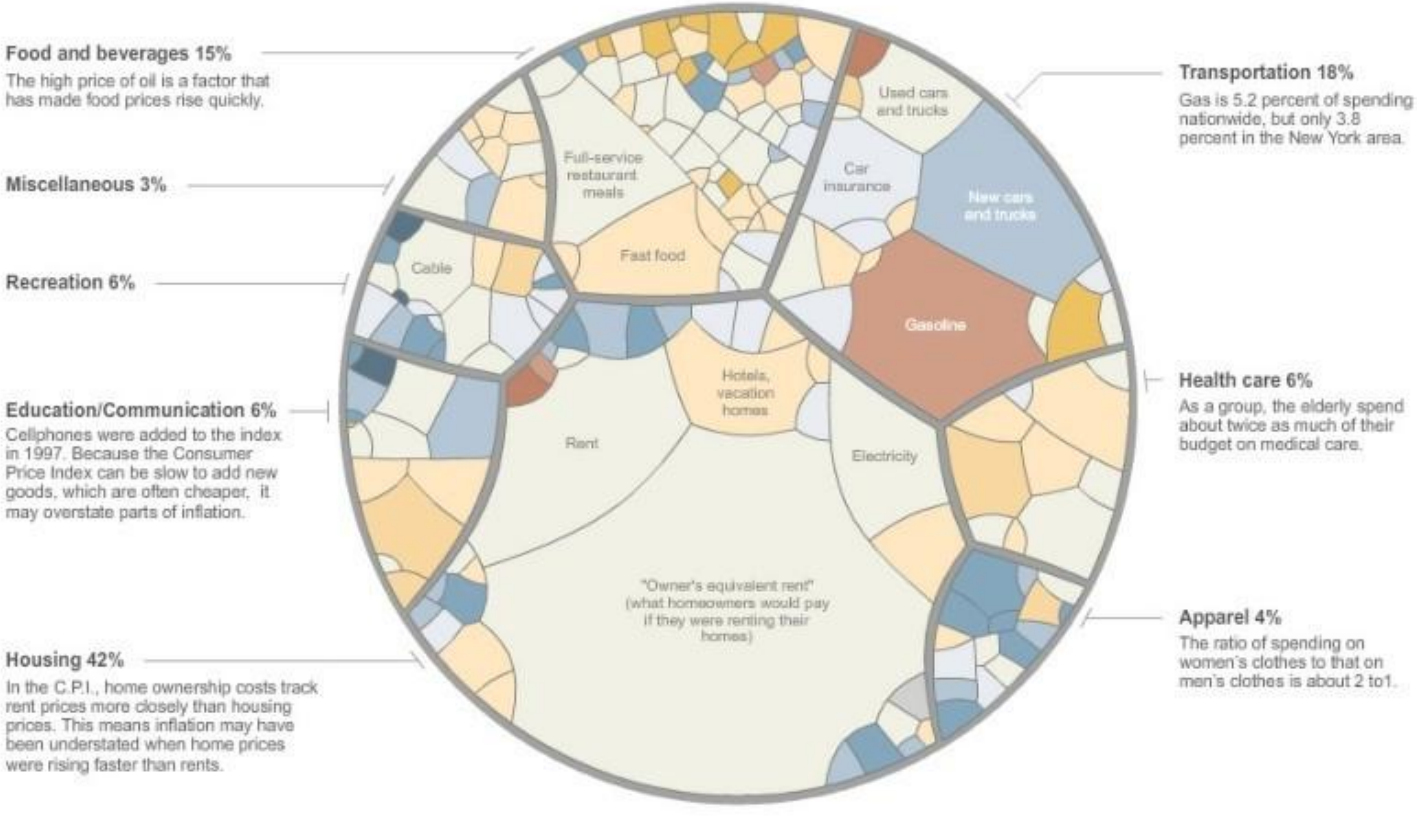


Spiral

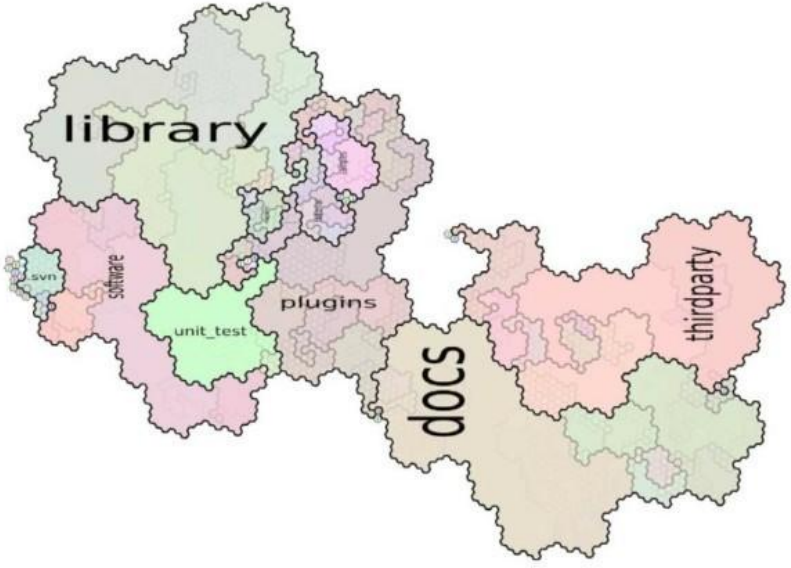


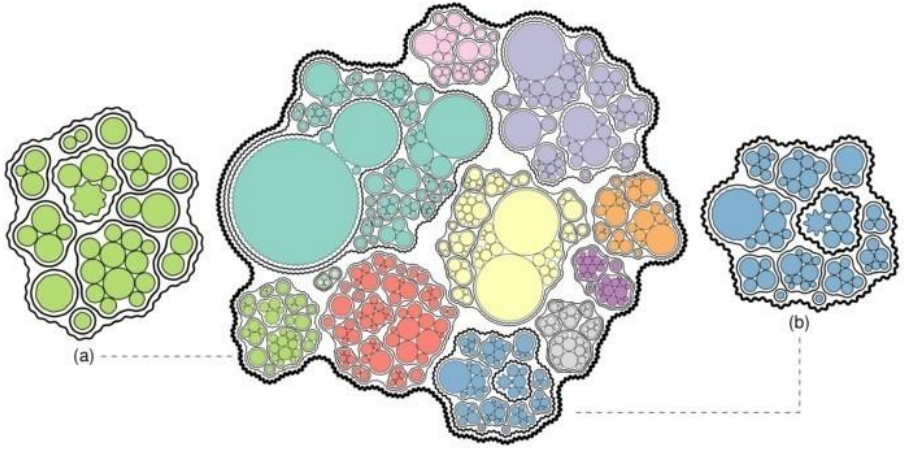
Ordered Squarified

空间的形状的变化


Food and beverages 15%
The high price of oil is a factor that has made food prices rise quickly.
Miscellaneous 3%
Recreation 6%
Education/Communication 6%
Cellphones were added to the index in 1997. Because the Consumer Price Index can be slow to add new goods, which are often cheaper, it may overstate parts of inflation.
Housing 42%
In the C.P.I., home ownership costs track rent prices more closely than housing prices. This means inflation may have been understated when home prices were rising faster than rents.
Transportation 18%
Gas is 5.2 percent of spending nationwide, but only 3.8 percent in the New York area.
Health care 6%
As a group, the elderly spend about twice as much of their budget on medical care.
Apparel 4%
The ratio of spending on women's clothes to that on men's clothes is about 2 to1.
Full-service restaurant meals
Used cars and trucks
Car insurance
New cars and trucks
Fast food
Cable
Gasoline
Hotels, vacation homes
Rent
Electricity
"Owner's equivalent rent" (what homeowners would pay if they were renting their homes)

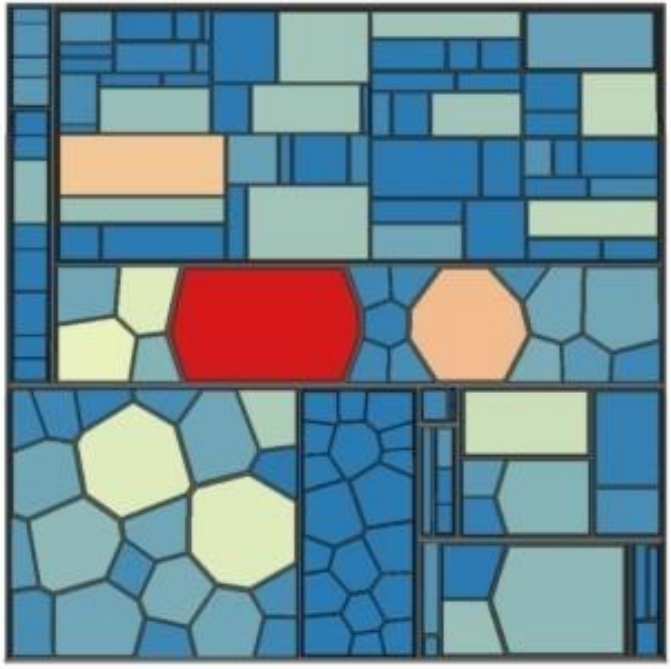
空间的形状的变化





混合树图布局算法

为单个树图可视化中的给定数据集的每个子层次结构选择和应用多种不同树图布局



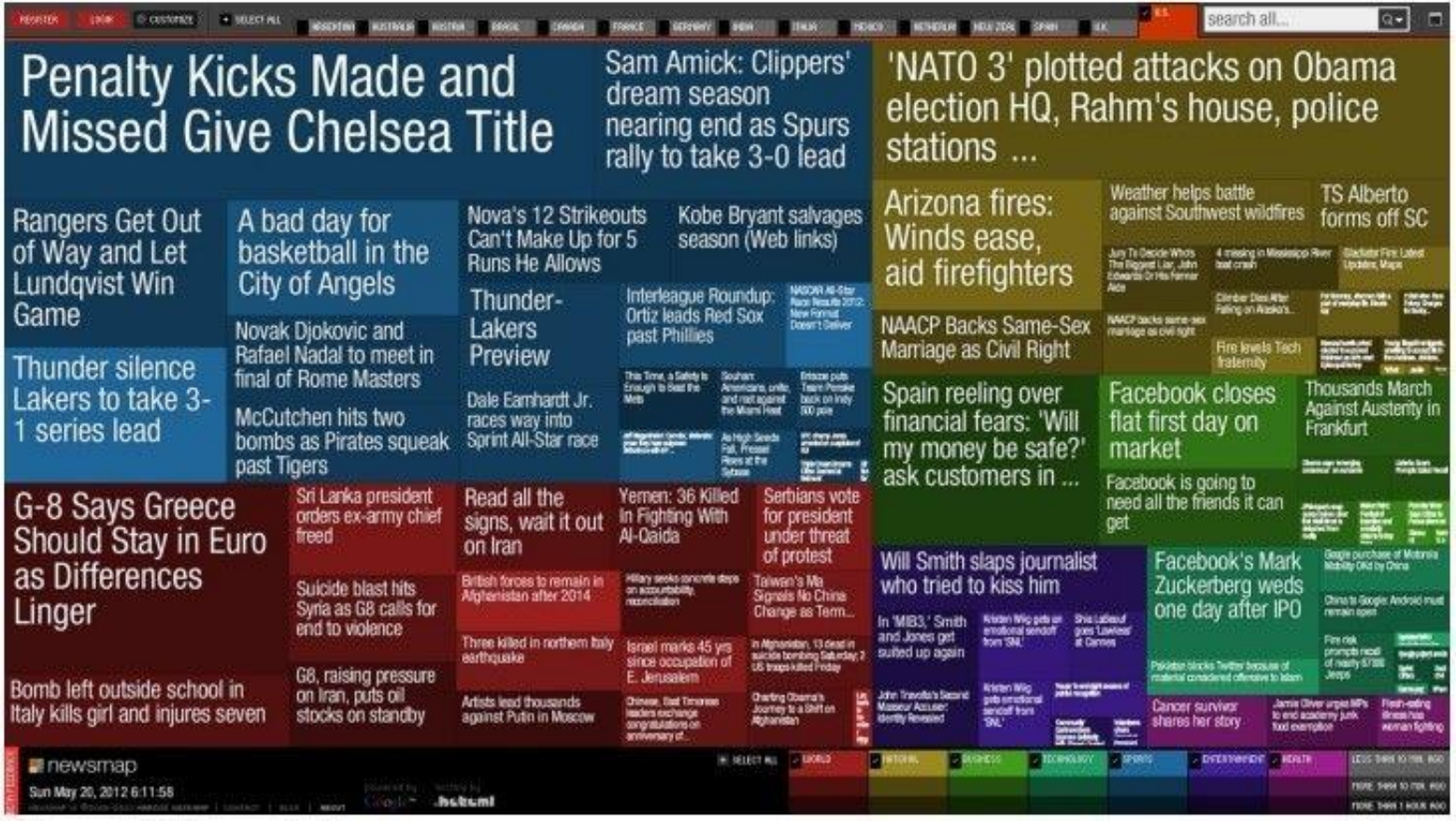
树图的交互

- 几种交互的功能和效果如下：
- (1)选中树图中的特定节点，由于父子节点的矩形相重叠，需要判断所选中节点的层次
- (2)选中节点后，改变其权值，重新生成树图
- (3)选中节点后，赋值或改变矩形的颜色
- (4)选中某节点后，放大显示以此节点为根节点的子树
- (5)层次下行操作的反向，显示以当前节点的父节点为根节点的子树
- (6)改变布局算法，重新生成树图
- (7)对使用者感兴趣的一个或多个子节点进行放大，再覆盖显示在全局的树图上方
- (8)以生成的全局树图的图像作为输入，应用图像处理技术中的变换方法，生成整体结构不变但部分区域得到平滑放大的新树图
- (9)允许在放大Slice-and-Dice Treemap中感兴趣的节点时，有限度地压缩其他节点
- (10)将关注的节点与周围节点的关系用弹簧模型进行建模，通过求解弹簧平衡的线性方程组来均匀放大多个关注的节点，压缩非关注的节点

树图交互分类

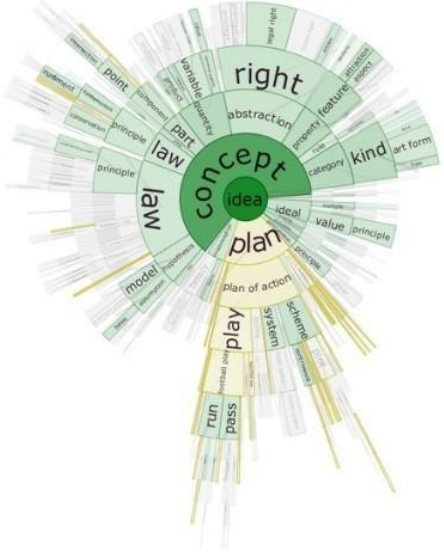
类　别	交　互
基础交互	节点选择
	权重改变
	颜色赋值与改变
	向下钻取
	向上钻取
	布局算法改变
创新交互	MagicLens
	鱼眼放大
	语义缩放
	气球焦点


Help
The Market
DJIA 12369.38
Nasdaq 2778.79
9:05 am May 20
Health Care
Financials
Oil & Gas
Utilities
Basic Materials
Consumer Goods
Technology
Telecommunications
Consumer Services
Apple Inc. +0.04%
AAPL last: $530.38 chg: +50.26
(click for more details)
Sector view
Color key (% change)
-6.0%
0
+6.0%
News
Icons
None
Show Change since
Close
26 weeks
52 weeks
YTD
Highlight Top 5
Gainers
Losers
None
Find (name or ticker)
Color scheme
red/green
blue/yellow


search all...
Penalty Kicks Made and Missed Give Chelsea Title
Sam Amick: Clippers' dream season nearing end as Spurs rally to take 3-0 lead
'NATO 3' plotted attacks on Obama election HQ, Rahm's house, police stations ...
Rangers Get Out of Way and Let Lundqvist Win Game
A bad day for basketball in the City of Angels
Nova's 12 Strikeouts Can't Make Up for 5 Runs He Allows
Kobe Bryant salvages season (Web links)
Arizona fires: Winds ease, aid firefighters
Weather helps battle against Southwest wildfires
TS Alberto forms off SC
Thunder silence Lakers to take 3-1 series lead
Novak Djokovic and Rafael Nadal to meet in final of Rome Masters
McCutchen hits two bombs as Pirates squeak past Tigers
Thunder-Lakers Preview
Dale Earnhardt Jr. races way into Sprint All-Star race
Interleague Roundup: Ortiz leads Red Sox past Phillies
NAACP Backs Same-Sex Marriage as Civil Right
Fire levels Tech fraternity
Spain reeling over financial fears: 'Will my money be safe?' ask customers in ...
Facebook closes flat first day on market
Facebook is going to need all the friends it can get
Thousands March Against Austerity in Frankfurt
G-8 Says Greece Should Stay in Euro as Differences Linger
Sri Lanka president orders ex-army chief freed
Suicide blast hits Syria as G8 calls for end to violence
G8, raising pressure on Iran, puts oil stocks on standby
Bomb left outside school in Italy kills girl and injures seven
Read all the signs, wait it out on Iran
British forces to remain in Afghanistan after 2014
Three killed in northern Italy earthquake
Artists lead thousands against Putin in Moscow
Yemen: 36 Killed In Fighting With Al-Qaida
Israel marks 45 yrs since occupation of E. Jerusalem
Serbians vote for president under threat of protest
Taiwan's Ma Signals No China Change as Term...
Will Smith slaps journalist who tried to kiss him
In 'MIB3,' Smith and Jones get suited up again
Facebook's Mark Zuckerberg weds one day after IPO
Google purchase of Motorola Mobility OKd by China
China to Google: Android must remain open
Cancer survivor shares her story
Jamie Oliver urges MPs to end academy junk food exemption
newsmap
Sun May 20, 2012 6:11:58

径向布局

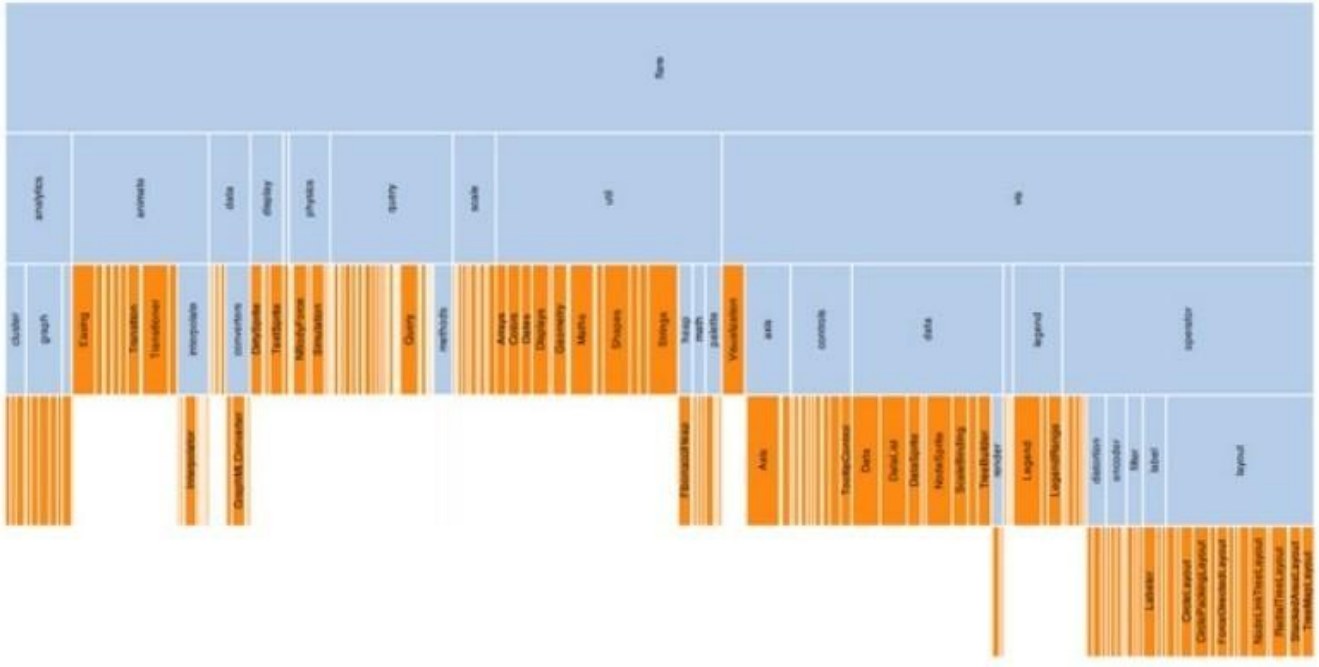
- 径向树图(Polar Treemap)
- 旭日图(Sunburst)
  - 中心的圆代表根节点，各个层次用同心圆环表示。圆环的划分依赖于相应层次上的节点数目及相关属性
  - 旭日图与树图法的不同之处在于，中间层次的节点可以被显示，因此容易分辨层次结构
  - 它的空间利用率比节点-链接法好，但弱于树图法
  - 当树结构不平衡的时候(如某棵子树层次深)，导致某一部分的扇形向外延伸很长，造成不合理的长宽比



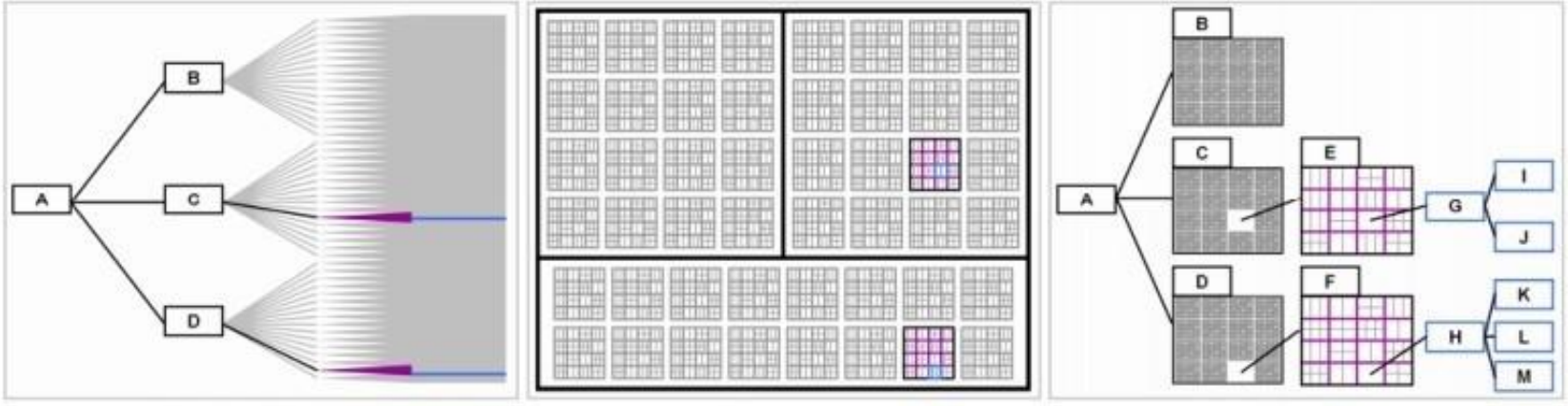
其他方法

- 节点-链接法和空间填充法各有优缺点：
  - 节点-链接法能清晰、直观地显示层次结构
  - 空间填充法能有效地利用空间，从而支持大规模的层次数据
  - 将两者组合，可结合双方的优势
- 相邻层次图(Adjacency Diagram)是节点-链接法的空间填充变
  - 采用填充的区域(弧或柱状区域)表达节点，相邻节点之间的位置关系则编码了彼此之间的层次关系

相邻层次图(Adjacency Diagram)

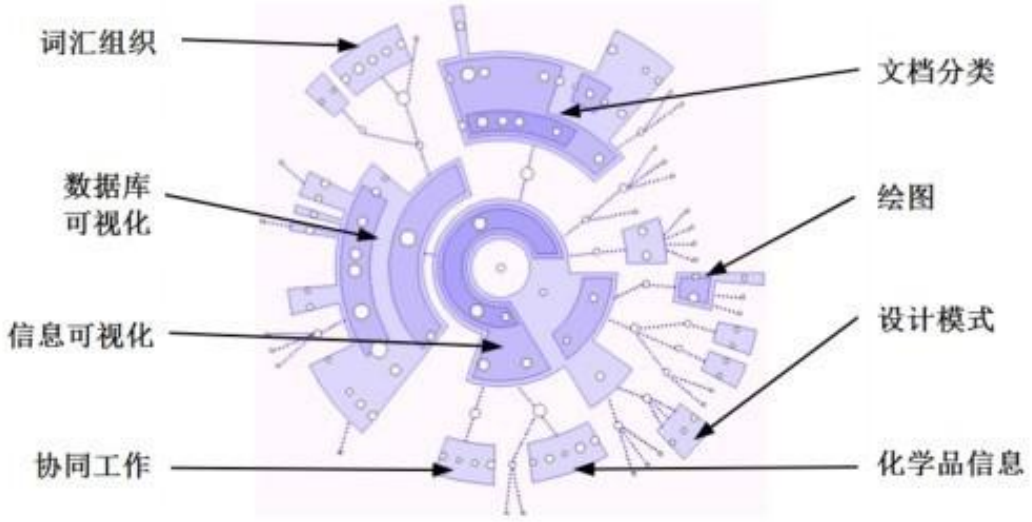


弹性层次法



Node–Link Diagram　　Treemap　　Elastic Hierarchy

混合式旭日图



2 网络数据

网络和图

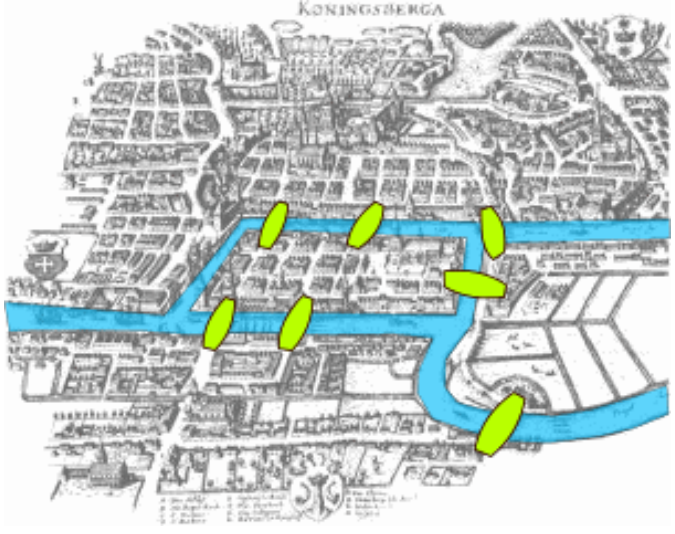
- 不具备层次结构的关系数据，可统称为网络(Network)数据
- 络通常用图(Graph)表示
  - 图G由一个有穷顶点集合V和一个边集合E组成。
  - 在图结构中，将节点称为顶点，边是顶点的有序偶对
  - 若两个顶点之间存在一条边，就表示这两个顶点具有相邻关系

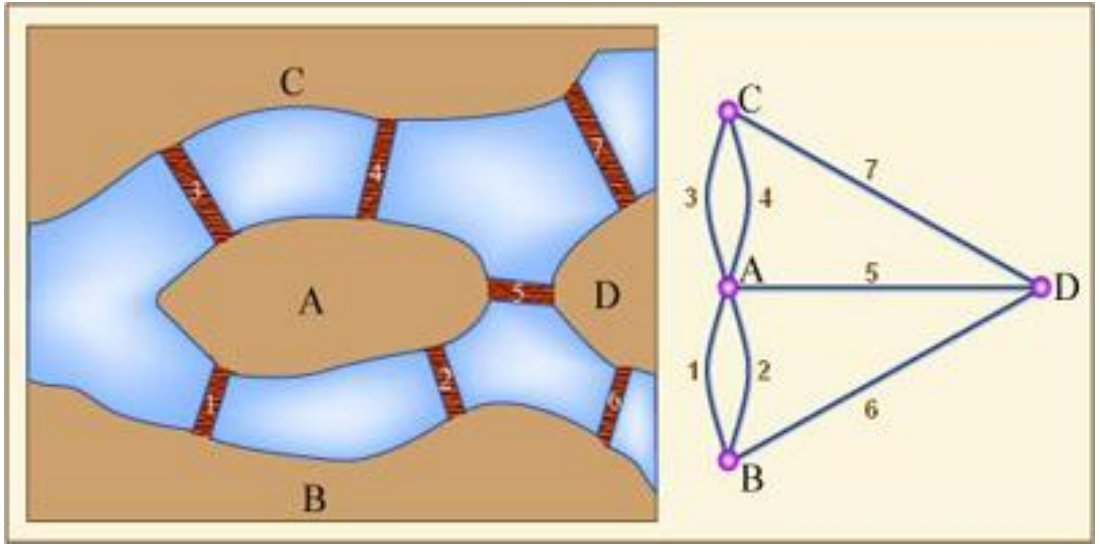
- 如每条边都定义了权重，则称为加权图。如果图的每条边都有方向，则称为有向图，否则是无向图

网络关系数据

- 网络数据并不具有自底向上或自顶向下的层次结构，表达的关系更加自由和复杂
  - 社交网络
  - 电话网络
  - 邮件网络
  - 合作网络

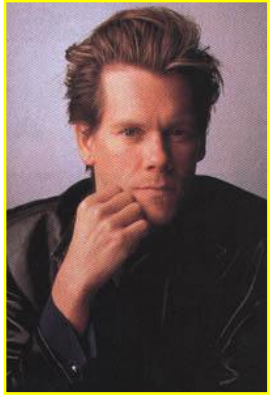
七孔问题

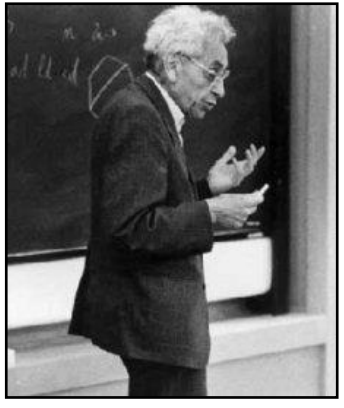




网络理论的应用

- 疾病传播分析
- 路由器网络的设计
- 搜索引擎的设计
- 演员的协作关系分析
- 科研人员的研究协作分析
- 社交网络



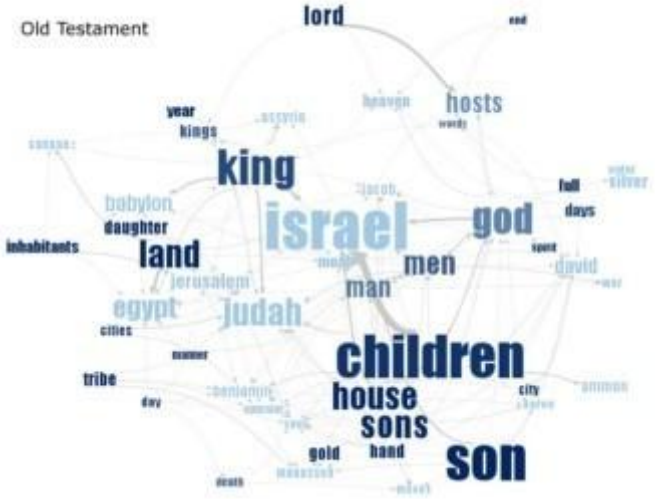


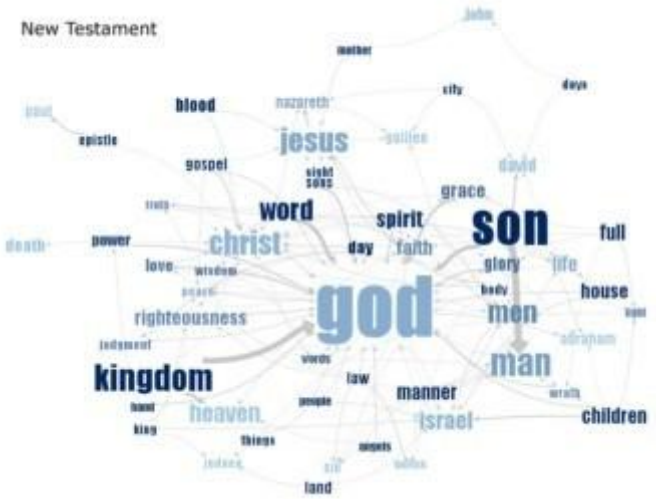
网络的重要性质

- 关系的复杂性
  - 和层次数据相比，关系可以存在于任意两个节点中
  - 关系可能具有方向性、权重
- 关系的中心性(centrality)
  - 度(degree), 相近/整体(closeness),
  - 中间(betweenness), 特征向量(eigenvector)

网络数据可视化和分析

- 揭示数据背后所隐藏的模式，帮助把握整体状况，协助管理和决策





网络数据可视化

- 图的绘制(Graph Drawing)，历史悠久。包括：
  - 网络布局
  - 网络属性可视化
  - 用户交互
  - 其中布局确定图的结构关系，是最核心要素

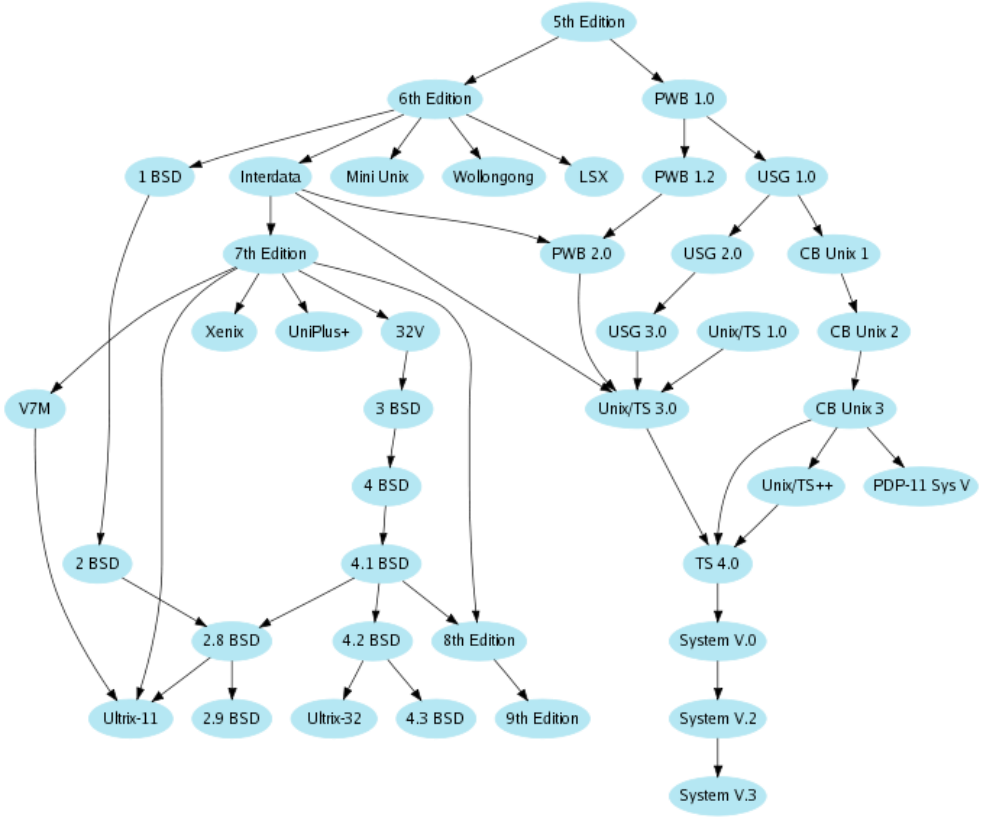
- 最常用的网络数据的布局
  - 节点-链接法
  - 相邻矩阵
  - 两者之间没有绝对的优劣，在实际应用中针对不同的数据特征以及可视化需求选择不同的可视化表达方式，或采用混合表达方式

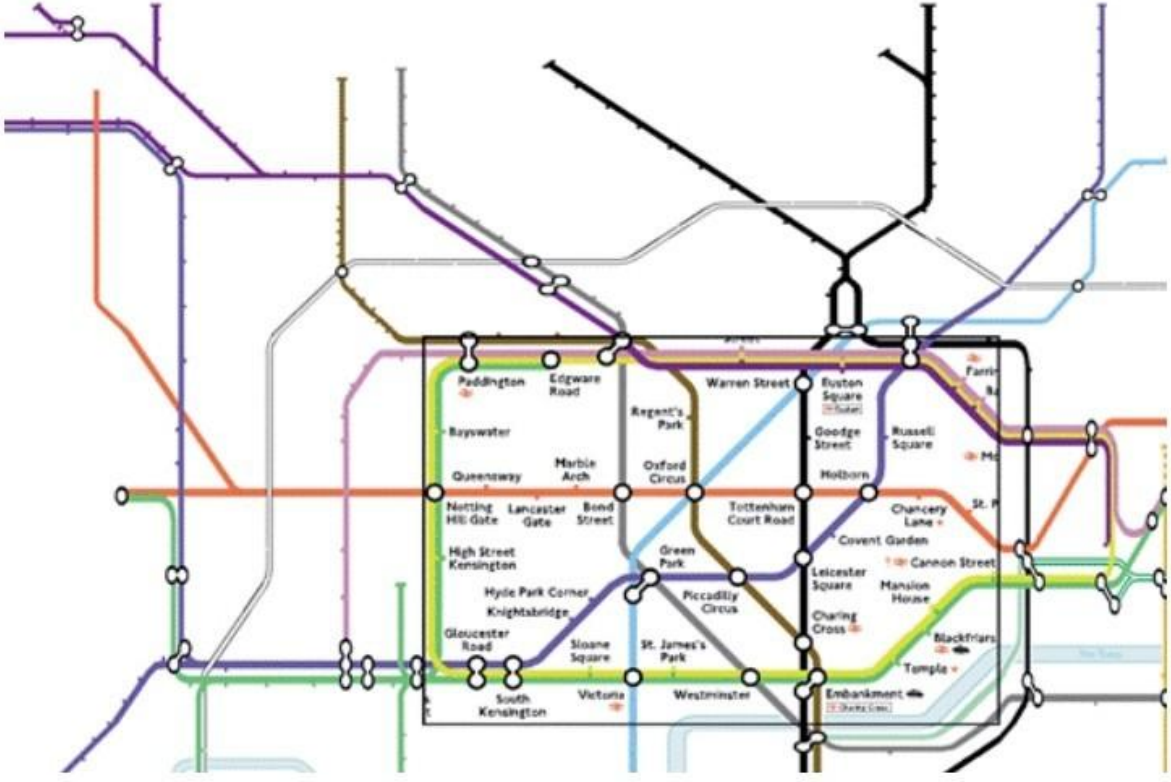
图的显示

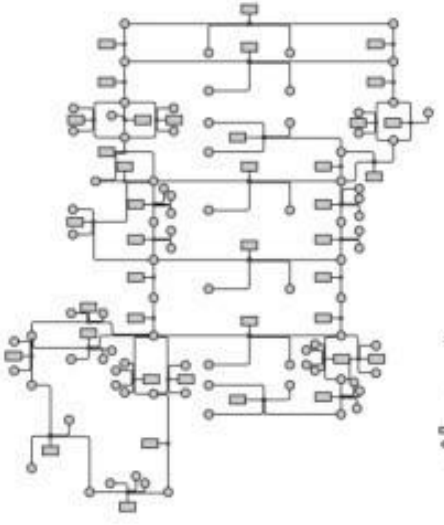
- 节点链接式显示
  - 分层显示/Sugiyama
  - 力导向布局
  - 多维尺度分析(Multi-Dimensional Scaling, MDS)布局
- 相邻矩阵
- 基于属性的显示

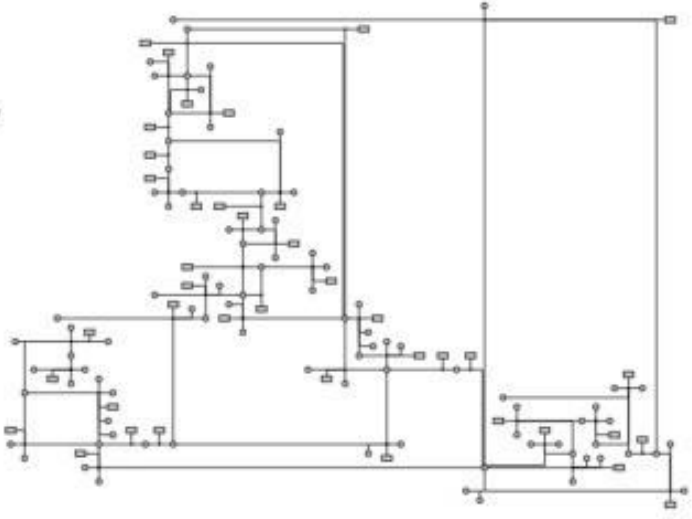
Sugiyama类显示

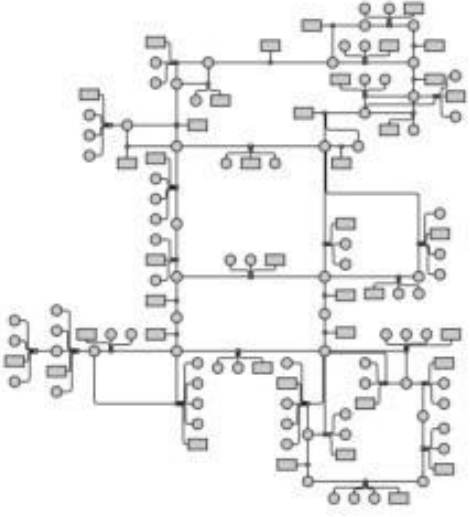
- 非常适用于显示具有原生顺序的树
- 图的“深度”映射到某一坐标轴上

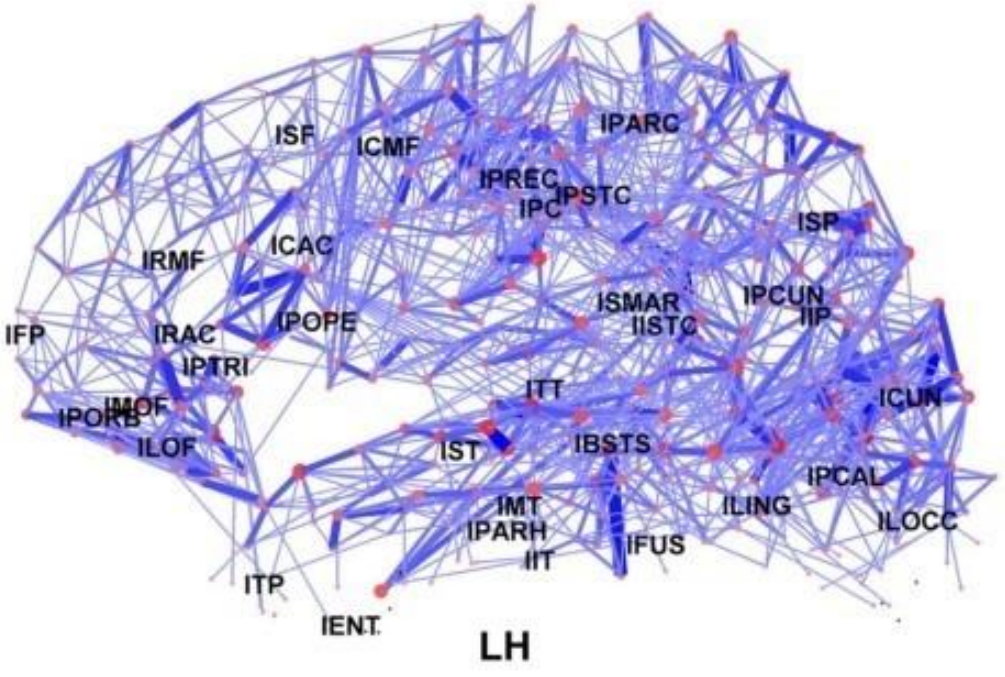



Paddington
Edgware Road
Warren Street
Euston Square
Bayswater
Regent's Park
Goodge Street
Russell Square
Queensway
Marble Arch
Oxford Circus
Holborn
Notting Hill Gate
Lancaster Gate
Bond Street
Tottenham Court Road
Chancery Lane
Covent Garden
High Street Kensington
Green Park
Cannon Street
Hyde Park Corner
Leicester Square
Mansion House
Knightsbridge
Piccadilly Circus
Charing Cross
Gloucester Road
Sloane Square
St. James's Park
Blackfriars
Temple
South Kensington
Victoria
Westminster
Embankment








ISF
ICMF
IPARC
IPREC
IPC
IPSTC
ISP
IRMF
ICAC
ISMAR
IPCUN
IIP
IFP
IRAC
IPOPE
IISTC
IPTRI
ITT
ICUN
IMOF
IPORB
ILOF
IST
IBSTS
IPCAL
IMT
ILING
IPARH
ILOCC
IFUS
IIT
ITP
IENT
LH

节点-链接布局方法

- 节点-链接布局方法主要有
  - 力引导布局(Force-directed Layout)
  - 基于距离的多维尺度分析(MDS)

力引导布局(Force-directed Layout)

- “启发式画图算法”，1984，Peter Eades
  - 目的是减少布局中边的交叉，尽量保持边的长度一致
  - 用弹簧模拟两个点之间的关系，受到弹力的作用后，过近的点会被弹开而过远的点被拉近；通过不断的迭代，整个布局达到动态平衡，趋于稳定

- 力引导布局算法，1991，Fruch1991
  - 加入点之间的静电力，通过计算系统的总能量并使得能量最小化
  - 改进的模型称为能量模型，可看成弹簧模型的一般化

力引导布局算法

对于平面上的两个点i和j，用d(i,j)表示两个点的欧氏距离，$s(i,j)$表示弹簧的自然长度，k是弹力系数，r表示两点之间的静电力常数，w是两点之间的权重

弹簧模型

$$E_s = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2} k(d(i,j) - s(i,j))^2$$

能量模型

$$E = E_s + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{r w_i w_j}{d(i,j)^2}$$

在确定的初始条件下，通过反复迭代可达到整个模型的能量最小化

算法的时间复杂度是O(n^3),迭代次数与步长有关，一般认为是O(n)

每次迭代都要两两计算点之间的力和能量，复杂度是O(n^2)

力引导布局的伪代码

设置节点的初速度为 (0,0)
设置节点的初始位置为任意位置
DO
    总动能 E:= 0// 所有粒子的总动能为 0
    FOR 每个节点 i
        净力 f:= (0,0)
        FOR 除该节点外的每个节点j
            净力 f:= 净力 f +节点 j 对节点 i 的库仑力
        ENDFOR
        FOR 该节点上的每个弹簧
            净力 f:= 净力 f + 弹簧 s 对节点 i 的弹力
        ENDFOR
        // 如果没有阻尼衰减，整个系统将一直运动下去
        节点 i 的速度 := (节点 i 的速度 + 步长 * 净力)* 阻尼
        节点 i 的位置 :=节点 i 的位置+步长 节点 i 速度
        总动能 := 总动能 + 节点 i 的质量(节点 i 的速度)^2
    ENDFOR
WHILE 两次循环总动能差<某间值 " &//" 即节点已经停止移动

力引导布局算法工具包

- Tulip(http://tulip.labri.fr/)
- Prefuse(http://www.prefuse.org)
- Gephi(http://gephi.org)
- Protovis(http://mbostock.github.io/protov

力引导布局的改进

- 力引导布局只能达到局部优化，而不能达到全局优化，并且初始位置对最后优化结果的影响较大
- 力引导布局的众多改进算法主要针对效率的优化，优化思路也大致分为，减少迭代次数、降低每次迭代的时间复杂度两种
  - 例如，FADE算法 [Aaron2001]，利用Barnes-Hut四叉树分解大大降低了计算任意两个点之间万有引力的复杂度，将O(n2)的迭代时间复杂度降低到O(nlog(n))，是目前主流的力引导布局实现方法

多维尺度分析布局(MDS Layout)

- MDS布局针对高维数据，用降维方法将数据从高维空间降到低维空间，力求保持数据之间的相对位置不变，同时也保持布局效果的美观性

- 力引导布局方法的局部优化使得在局部点与点之间的距离能够比较忠实地表达内部关系，但却难以保持局部与局部之间的关系

- MDS是一种全局控制，目标是要保持整体的偏离最小

MDS

设V= $\{1,2,3,\cdots,n\}$是高维空间的n个数据点，矩阵D $\in R_{n\times n}$是数据点两两之间相异性相邻矩阵，即矩阵D中的每个d_i，表达了第i, j两个点之间的相异性$(i,j \in V)$,坐标矩阵X于所有点i, j都有:$\|x_i - x_j\| \approx d_{ij}$

求解上述问题有两种方法：一种是古典尺度分析方法，另一种就是基于距离的尺度分析方法

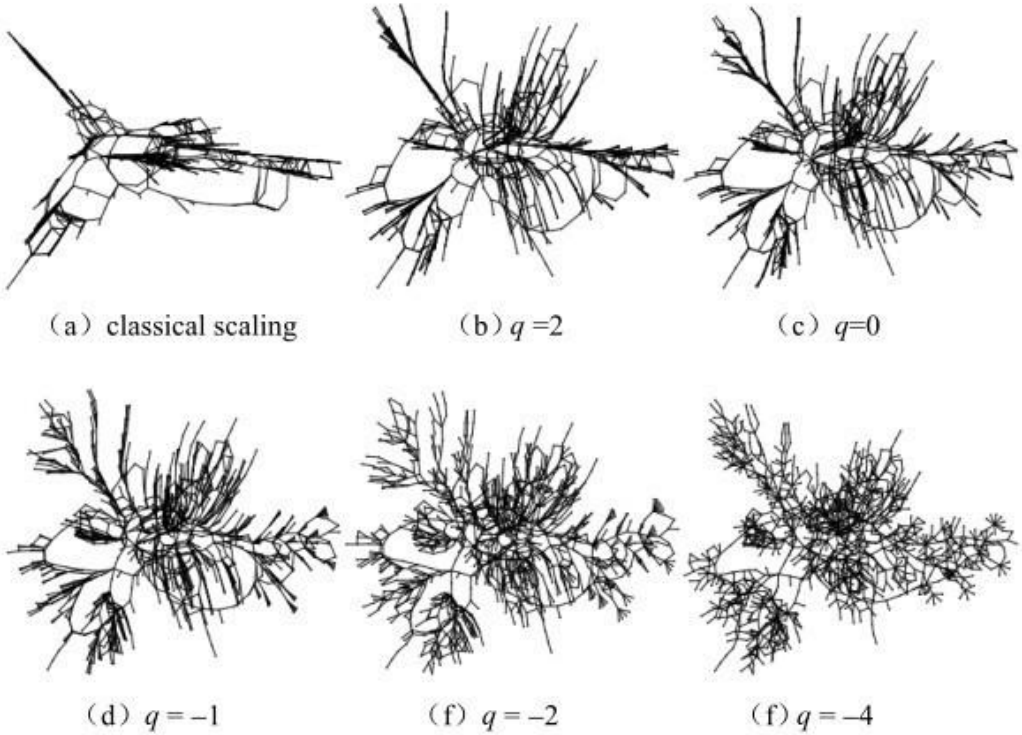
古典尺度分析基于矩阵近似和基本欧式几何理论，计算伪内积空间B与内积空间中点的相异性。这种借助内积空间迂回的求解方法大大增加了计算的空间复杂度和时间复杂度，对于数据规模的扩展有一定的局限性。而且求解过程有时会导致退化解，使输入不再对输出产生影响

基于距离的尺度分析方法的思想是使两点的距离尽量等价地表达它们的相异性，也就是求解一个优化问题，使高维距离和相异性差的误差函数Stress最小:

$$\text{Stress}(\boldsymbol{X}) = \sum_{i,j} w_{ij}\left(d_{ij} - ||x_i - x_j||\right)^2$$

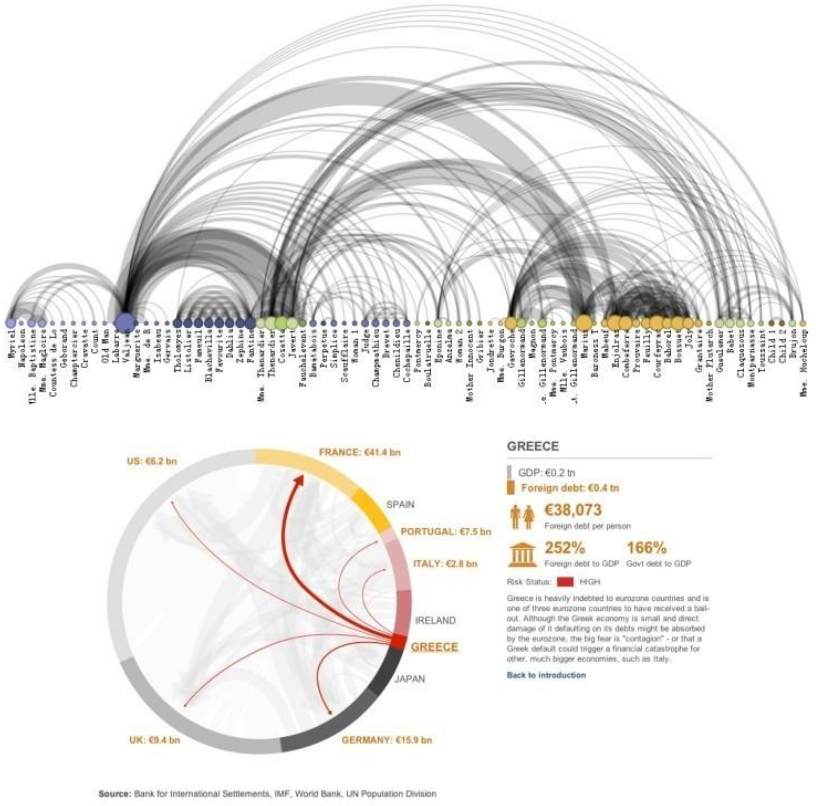
求解以上优化函数一般可采用梯度下降法，称为应力最大化

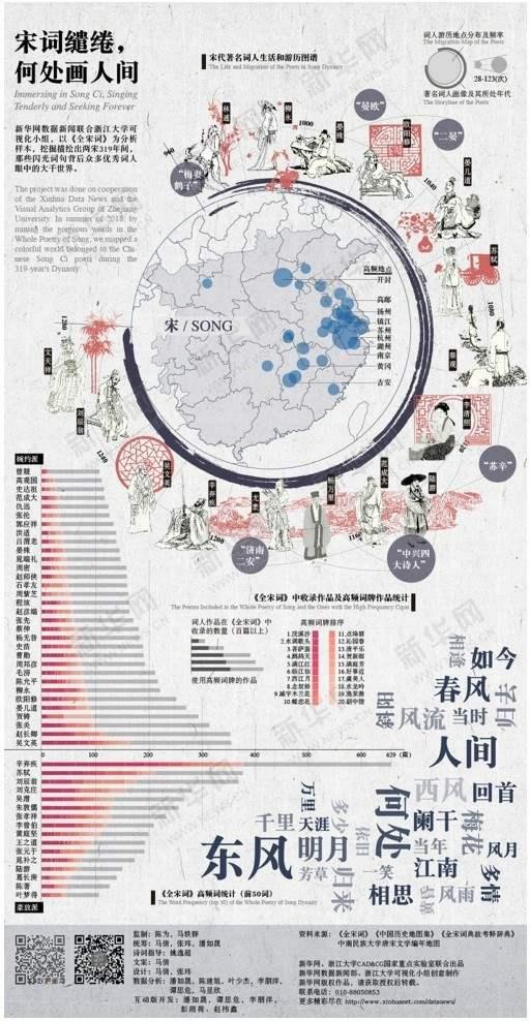
MDS



弧长链接图

- 弧长链接图(Arc Diagram)，采用一维布局方式，节点沿某个线性轴或环状排列，圆弧表达节点之间的链接关系
- 不能像二维布局那样表达图的全局结构，但在节点良好排序后可清晰地呈现环和桥的结构。
- 对节点的排序优化问题又称为序列化(Serialization)




宋词缱绻，何处画人间
Immersing in Song Ci, Singing Tenderly and Seeking Forever
新华网数据新闻联合浙江大学可视化小组，以《全宋词》为分析样本，挖掘描绘出两宋319年间，那些闪光词句背后众多优秀词人眼中的大千世界。
The project was done on cooperation of the Xinhua Data News and the Visual Analytics Group of Zhejiang University. In summer of 2018, by mining the gorgeous words in the Whole Poetry of Song, we mapped a colorful world belonged to the Chinese Song Ci poets during the 319-year's Dynasty
宋代著名词人生活和游历图谱
The Life and Migration of the Poets in Song Dynasty
词人游历地点分布及频率
The Migration Map of the Poets
28-123(次)
著名词人画像及其所处年代
The Storyline of the Poets
"晏欧"
"二晁"
"梅妻鹤子"
"苏辛"
"婉约派"
"豪放派"
"婉约派"
"豪放派"
高频地点
开封
高邮
扬州
镇江
苏州
杭州
湖州
南京
黄冈
吉安
宋 / SONG
"婉约派"
曾觌
高观国
李达昔
范成大
仇远
张抡
郭应祥
洪适
吕渭老
姜特
吴曾祺
周密
赵师侠
石孝友
周紫芝
程珌
赵彦瑞
张先
蔡伸
杨无咎
史浩
曾勤
周邦彦
毛滂
陈允平
柳永
欧阳修
晏几道
贺铸
张贲
赵长卿
吴文英
"豪放派"
辛弃疾
苏轼
刘辰翁
刘克庄
吴潜
朱敦儒
张孝祥
李曾伯
黄庭坚
王之道
张元干
晁补之
陆游
葛长庚
陈著
叶梦得
0
100
200
300
400
500
600
629（篇）
《全宋词》中收录作品及高频词牌作品统计
The Poems Included in the Whole Poetry of Song and the Ones with the High Frequency Cipai
词人作品在《全宋词》中收录的数量（百篇以上）
常用高频词牌的作品
高频词牌排序
1.浣溪沙
2.水调歌头
3.菩萨蛮
4.鹧鸪天
5.满江红
6.临江仙
7.西江月
8.念奴娇
9.减字木兰花
10.醉思凡
11.点绛唇
12.长相思
13.清平乐
14.贺新郎
15.满庭芳
16.好事近
17.虞美人
18.水龙吟
19.瑞龙吟
20.朝中措
如今
春风
相逢
团圆
当时
风流
人间
回首
西风
何处
阑干
梅花
风月
多情
当年
江南
依旧
一笑
相思
归来
芳草
明月
千里
天涯
万里
多少
东风
《全宋词》高频词统计（前50词）
The Word Frequency (top 50) of the Whole Poetry of Song Dynasty
监制：陈为、马修静
统筹：马倩、张玮、潘加晟
诗词指导：姚逸超
文案：马倩
设计：马倩、张玮
数据分析：潘加晟、陈建旭、叶少杰、李朝洋、谭思危、马星欣
互动版开发：潘加晟、谭思危、李朝洋、彭雨荷、赵伟鑫
资料来源：《全宋词》《中国历史地图集》《全宋词典故考释辞典》中南民族大学唐宋文学编年地图
新华网、浙江大学CAD&CG国家重点实验室联合出品
新华网数据新闻部、浙江大学可视化小组创意制作
新华网版权作品，请获取授权后转载。
联系电话：010-88050853
更多精彩尽在 http://www.xinhuanet.com/datanews/

相邻矩阵布局

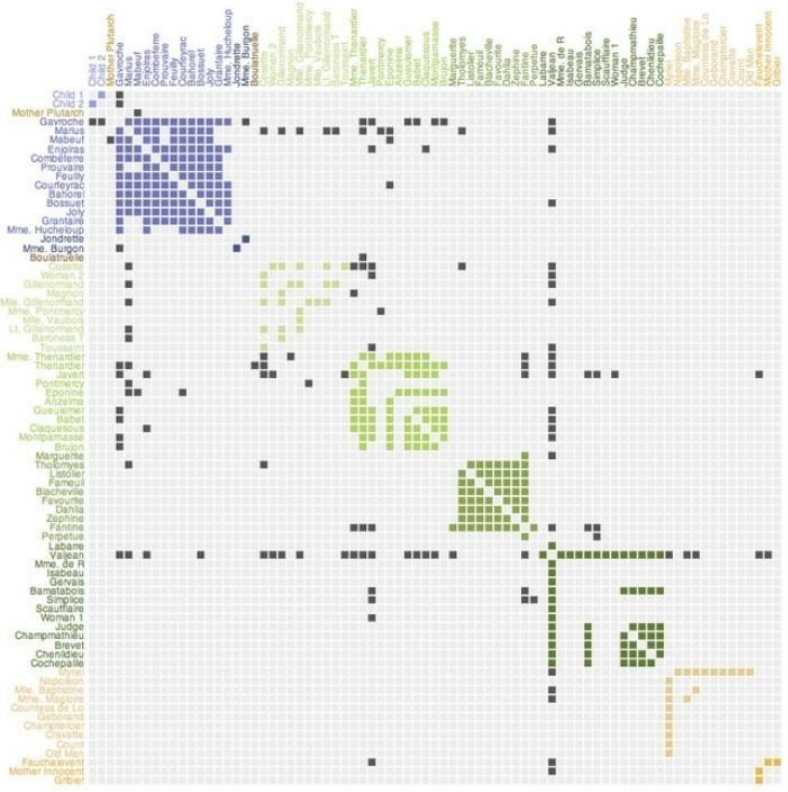
- 相邻矩阵(Adjacency Matrix)指代表N个节点之间关系的N×N的矩阵，矩阵内的位置(i,j)表达了第i个节点和第j个节点之间的关系
- 无权重的关系网络用01矩阵(Binary Matrix)来表达两个节点之间的关系是否存在
- 带权重的关系网络，相邻矩阵则可用(i,j)位置上的值代表其关系紧密程度
- 无向关系网络，相邻矩阵是一个对角线对称矩阵
- 有向关系网络，相邻矩阵不具对称性；相邻矩阵的对角线表达节点与自己的关系

相邻矩阵

- 相邻矩阵的表达简单易用
  - 可以用数值矩阵，也可以将数值映射到色彩空间表达
  - 但是，从相邻矩阵中挖掘出隐藏的信息并不容易，需要结合人机交互
  - 最关键的两种交互是排序和路径搜索，前者使具有相似模式的节点靠得更近，而后者则用于探索节点之间的传递关系
- 为相邻矩阵排序
  - 用于凸显网络关系中存在的模式

相邻矩阵

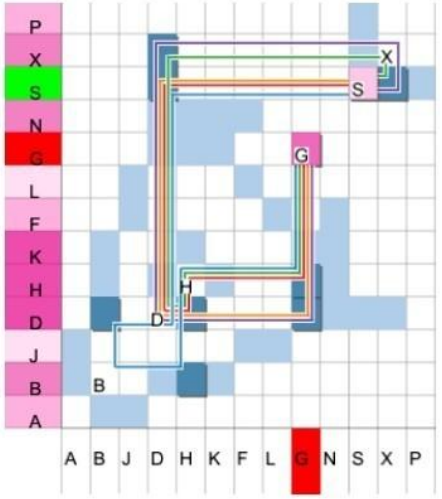
- 常规的排序方法依据网络数据的某一数值(矩阵值或节点的度)的大小执行
- 选择不同的排序项产生不同的矩阵排序结果

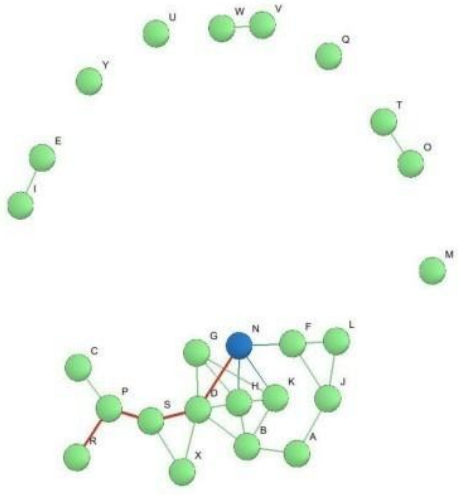


相邻矩阵

- 相邻矩阵法可显著表达节点之间的直接关系，而对间接关系，也就是关系传递性的可视表达比较薄弱。

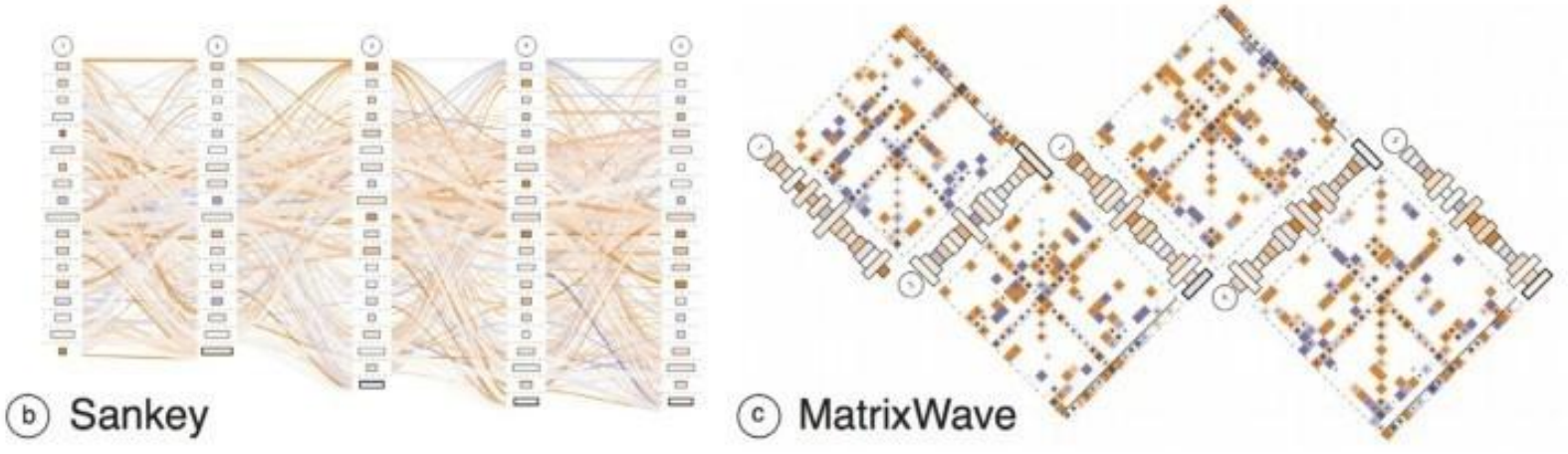
- 需要设计相邻矩阵的路径可视化算法，即可视地表达两个节点之间的最短路径





相邻矩阵

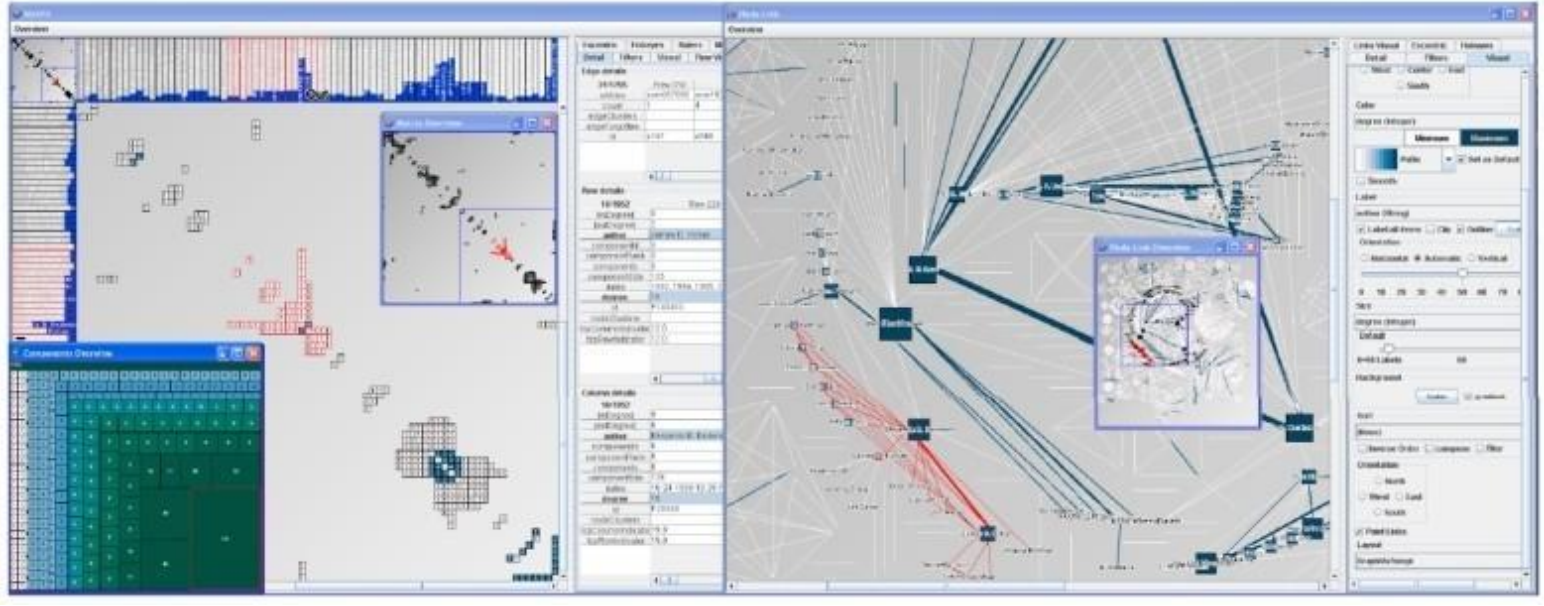
- 与节点-链接法相比，相邻矩阵最大的好处是可以完全规避边的交叉



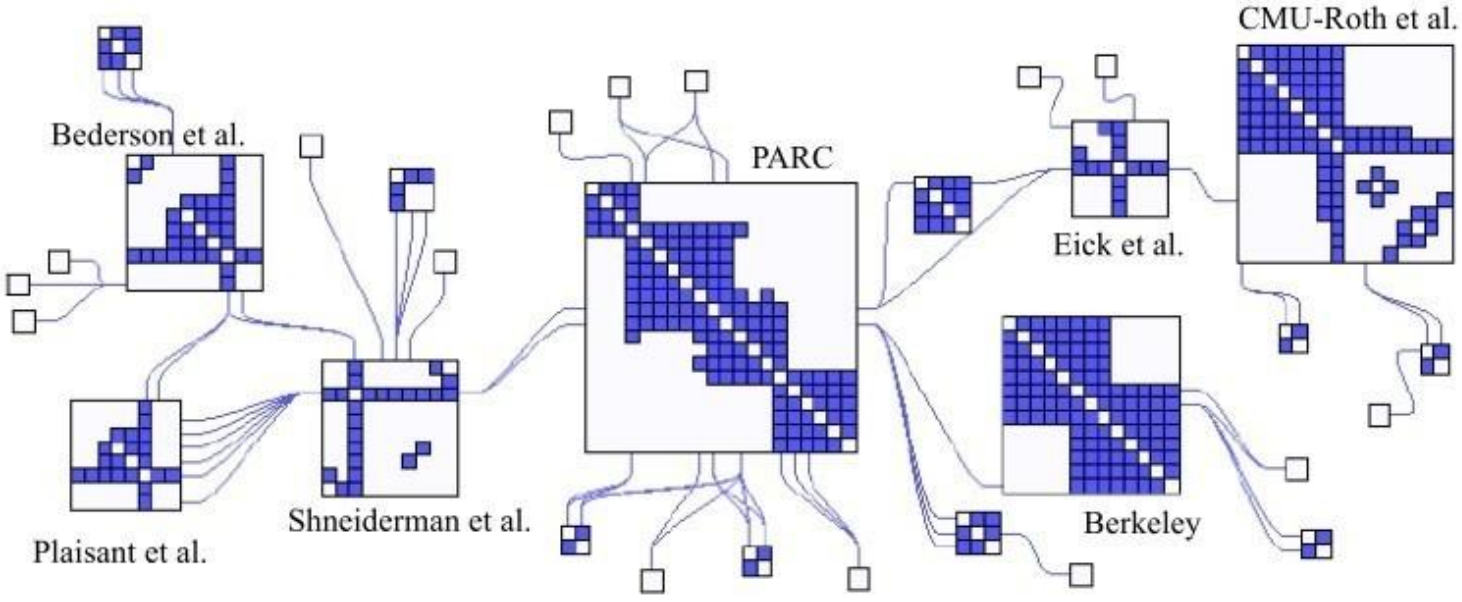
混合布局方法

- 节点-链接布局适用于节点规模大但边关系较为简单，并且能从布局中看出图的拓扑结构的网络数据
- 相邻矩阵适用于节点规模较小，但边关系复杂，甚至是两两节点之间都存在关系的数据
- 对于部分稀疏部分稠密的数据，可混合两者的布局设计
- 是否需要混合布局、如何设计混合布局必须经过仔细思考
  - 是否一种布局已经足以表达数据的模式？
  - 多种布局如何混合成一种布局？

MatrixExplorer方法



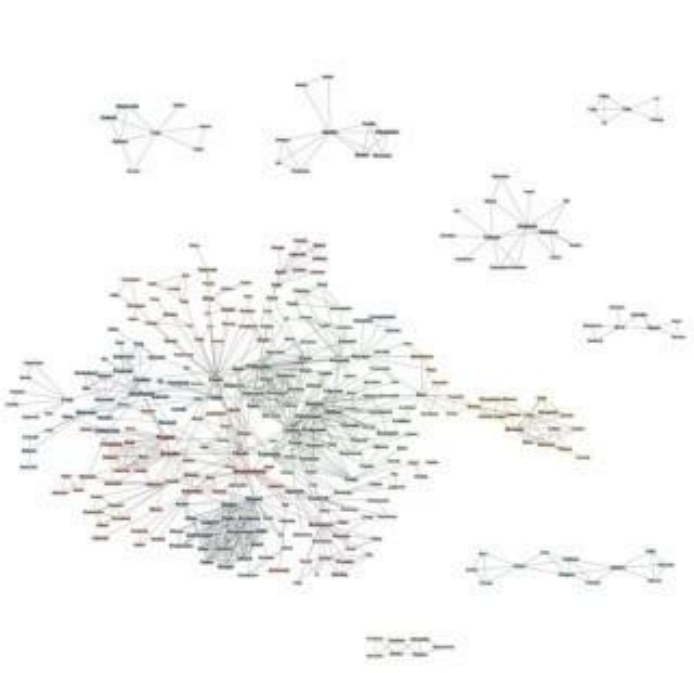
MatrixExplorer方法

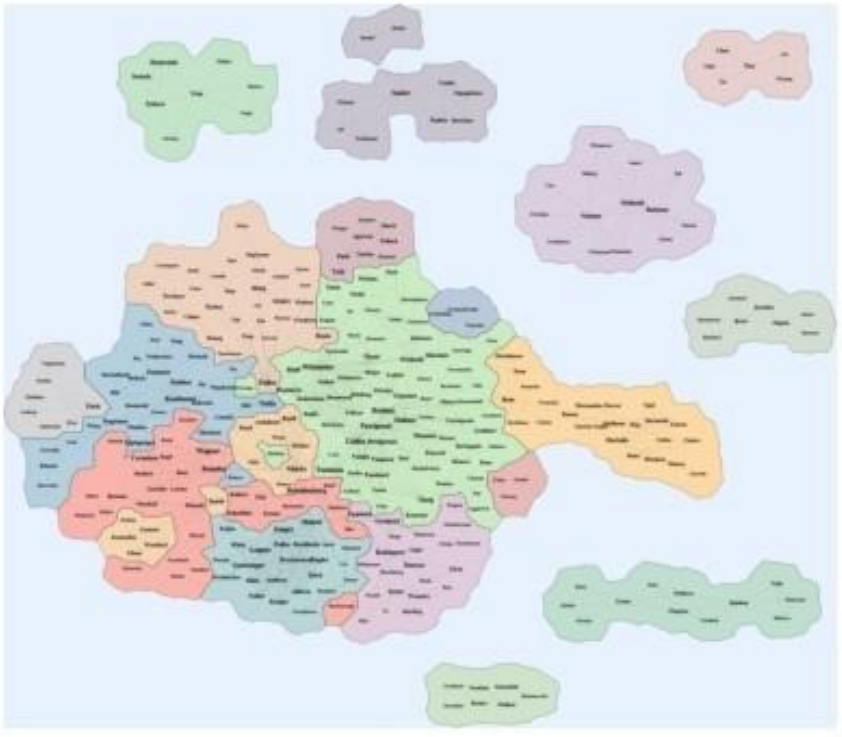


网络数据的地图隐喻可视化

- 将网络图用地图形式表达的方法，称为Gmap
- GMap是一种用平面代表集合，平面划分代表数据聚类的“地图”可视化策略，地图上的国家、国家之间的关系作为可视化隐喻表达了数据的分类和类别之间的相邻度关系
  - 将网络数据布置于二维空间
  - 用聚类分析的方法将网络图中的节点归类
  - 根据各个类别中点的分类情况构造Voronoi 图。一个Voronoi图代表地图的一个区域
  - 给地图的每个Voronoi区域上色

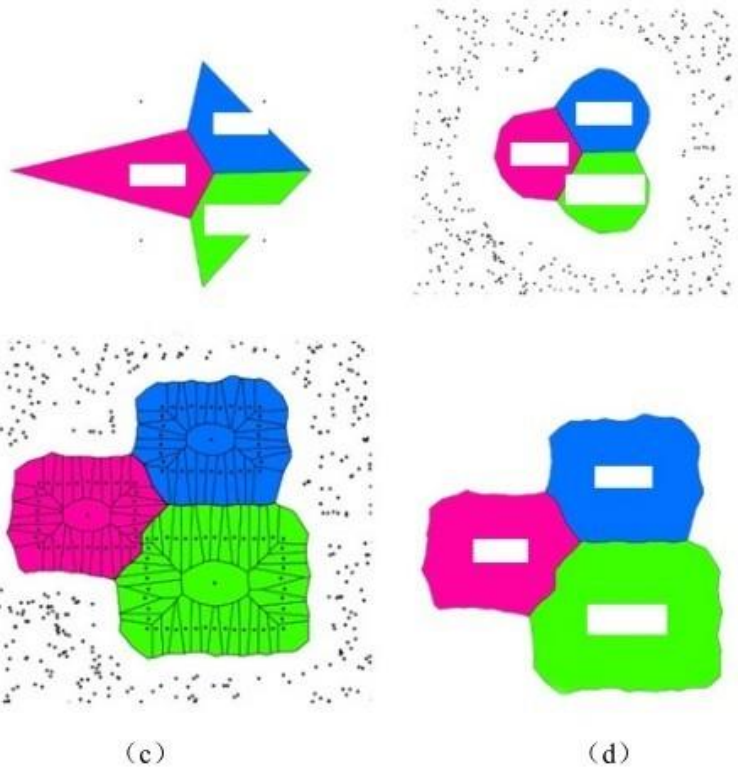
GMap



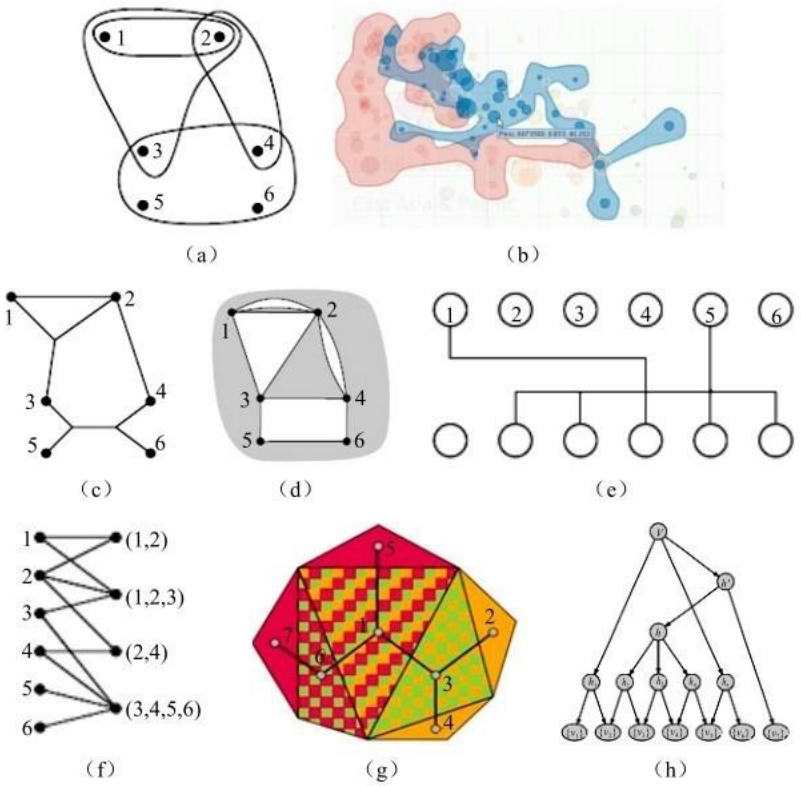


GMap

- 首先要通过降维方法将高维数据降成二维数据画在图上
  - 常见的降维方法有PCA、LLE、IsoMap、谱聚类，或力引导布局和多维标度布局方法
- 第2步的点聚类，也就是为了“划分势力”，定义每个集合中的点。
- 第3步GMap的关键步骤



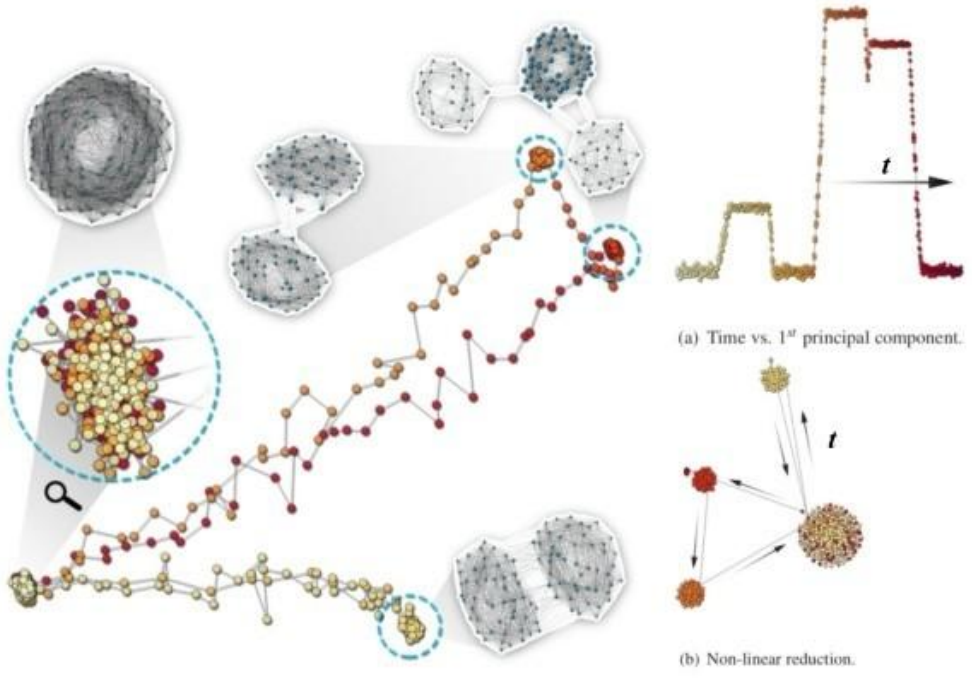
超图及其可视化



动态网络数据可视化

- 动态网络数据是流数据的一种，其中“动态”一词可以理解为节点的增减、关系的增减、节点/关系权重的变化三种
- 动画是一种直观的可视化技术，它主要基于点边图(Node Link Diagram)来表示
- 相比而言，时间轴技术不那么直观，但它更侧重于分析时变特性
  - 既可以使用点边图表示，也可以使用矩阵表示

[Elzen2016]提出了一种对动态网络进行二维投影的方法，将每个动态网络表示成一个静态的点边图



图可视化的视觉效果

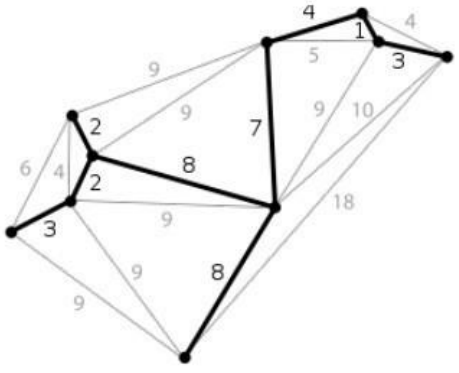
- 人类很早就意识到可以用圆圈表示节点、线段表示节点之间的连接关系这样的方式来对图进行可视化描述

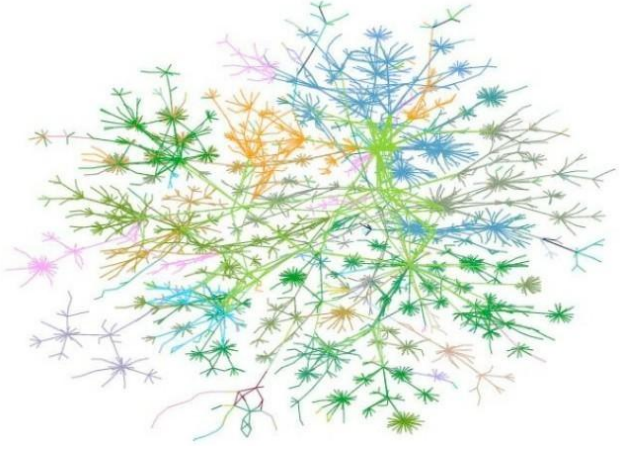
- 随着网络数据规模的不断扩大，传统方法绘制的结果中，节点和边经常出现互相遮挡，形成极高的视觉混杂度(Visual Clutter)，甚至会阻碍我们对真实数据的认知。

- 21世纪初起，在信息可视化和图绘制两个科学研究领域，分别涌现出大量的成果来解决这些问题
  - 一种思路是根据信息可视化的信息分级(Level of Detail)原则，对大规模图进行层次化简化
  - 另一种思路是在尽量不减少原图信息量(包括边和节点的数目)的前提下，对图进行基于骨架的聚类。无论采取哪种思路，其目的都是为了应对大规模图对有限可视化空间的挑战，降低网络数据可视化的视觉混杂度，挖掘和展示数据背后隐藏的信息

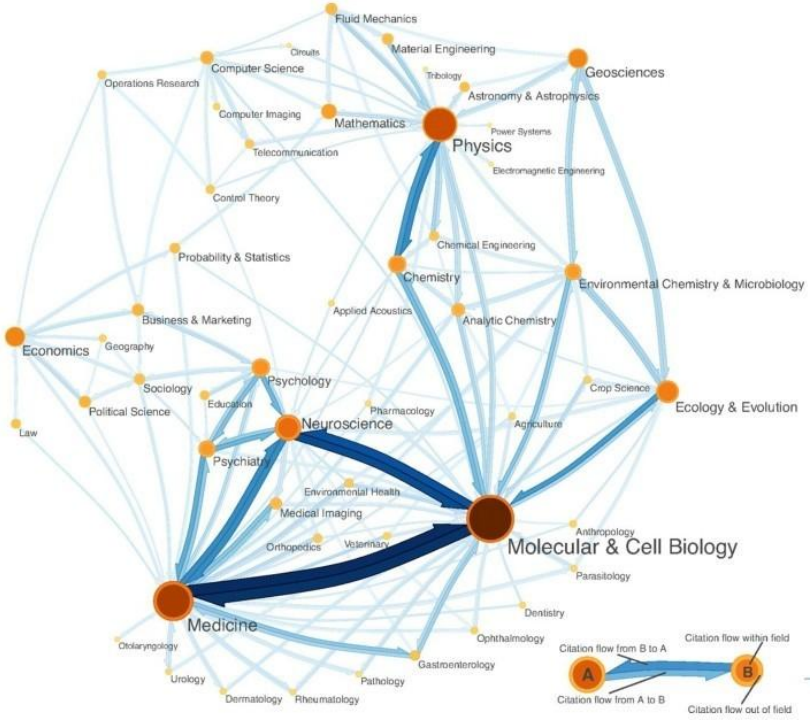
图的拓扑简化

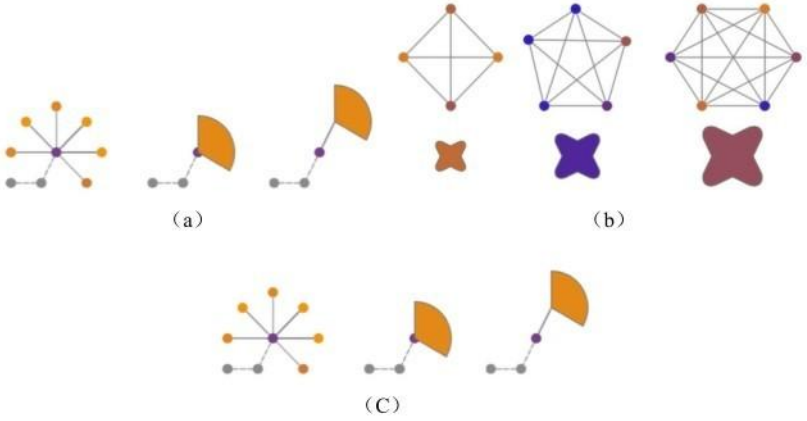
- 图的拓扑结构包括两个部分：节点和边。对应的，对图的拓扑进行简化也存在两种方法，即分别对节点和边进行层次化简化
  - 对边的提取，如最小生成树
  - 另一种方法是将强连通的节点进行聚类，并把聚类后的节点集作为一个新的超级节点绘制到可视化结果中

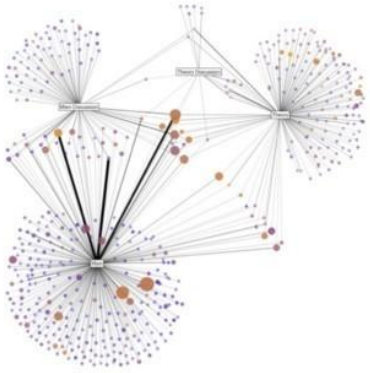


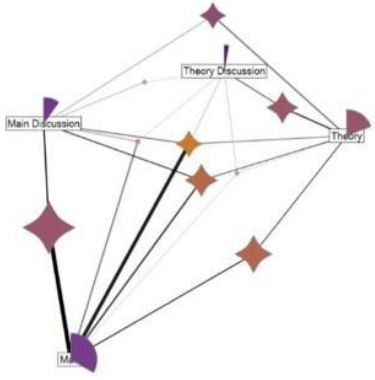


- [Dun2013]提出了一种图形简化法，这种方法把经常出现的具有代表性的点边模式表示成紧密的有意义的图形



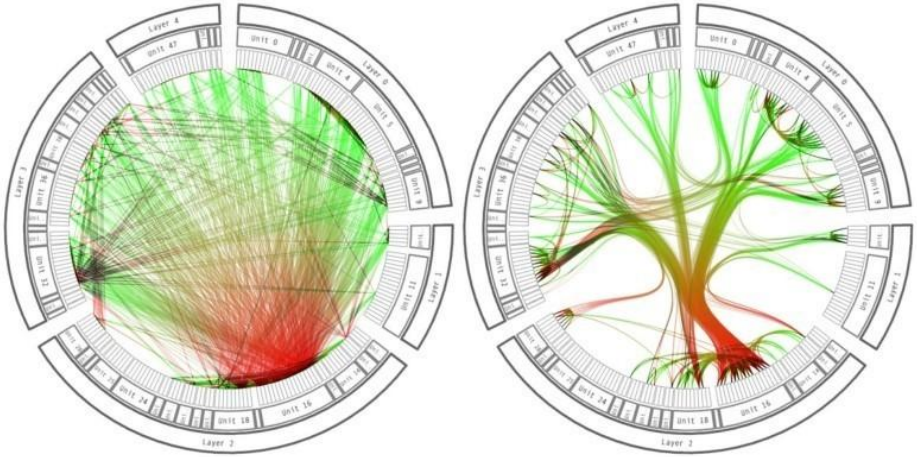

（a）
（b）
（C）




Theory Discussion
Main Discussion

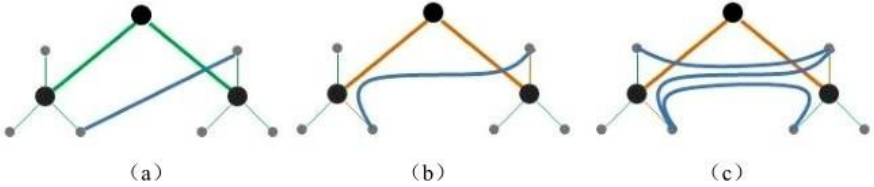
图的边绑定

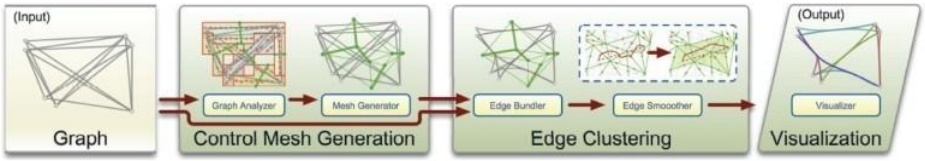
- 拓扑简化方法
  - 优势：在前端的数据处理阶段就减少图的复杂程度，使得在可视化阶段可以直接绘制简化后的图
  - 副作用：带来信息的丢失；在获得更高层次数据抽象结果的同时，丢失细节信息
- 边绑定，核心思想：在保持信息量(即不减少边和节点总数)的情况下，将图上互相靠近的边捆绑成束，从而达到去繁就简的效果



边绑定技术

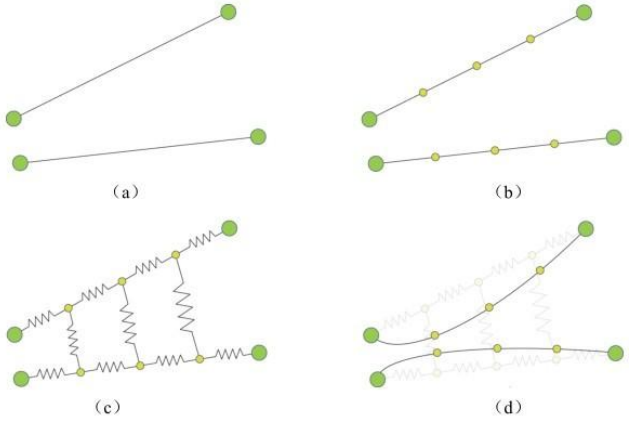
- 边绑定通过重新排列边的路径，将地理上或者语义上相近的边捆绑在一起来降低图中的视觉混乱
- HEB算法仅仅适用于具有层次化结构的图，更多的研究者开始关注如何对普通图的边进行绑定的技术
- 基于几何的边绑定方法(GBEB)[Cui2008]是第一个成功对普通图进行边绑定的算法



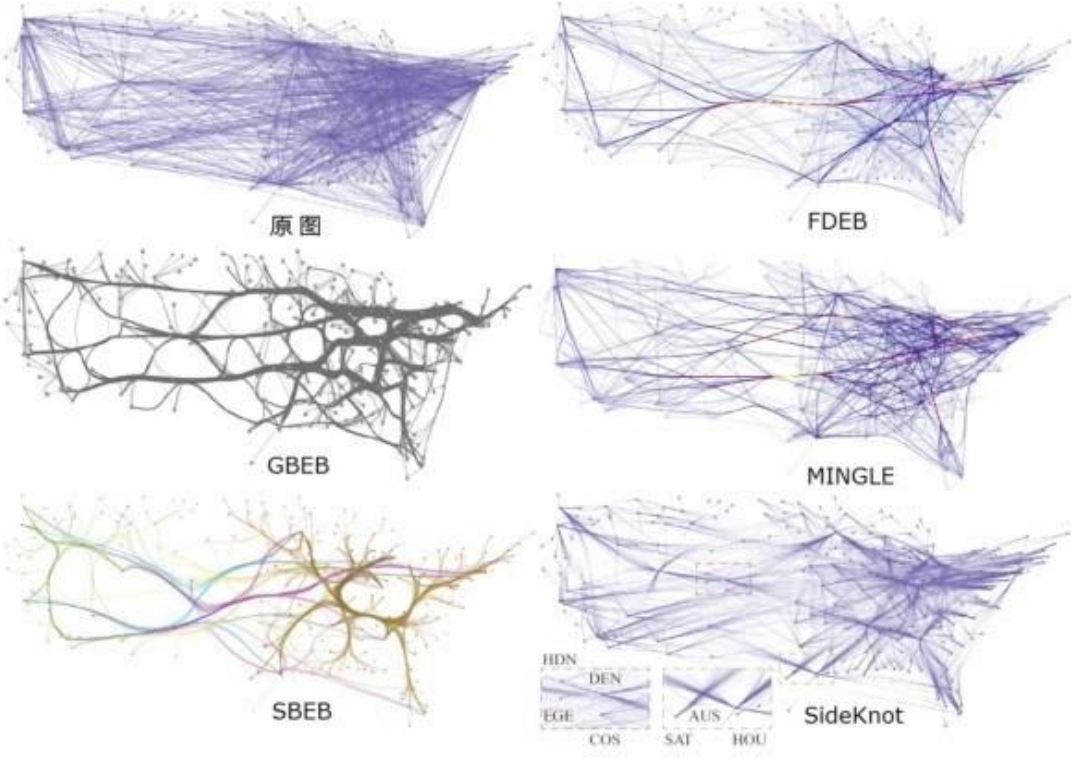


边绑定技术

- 基于力引导的边绑定技术(FDEB算法)
- SideKnot算法



边绑定技术



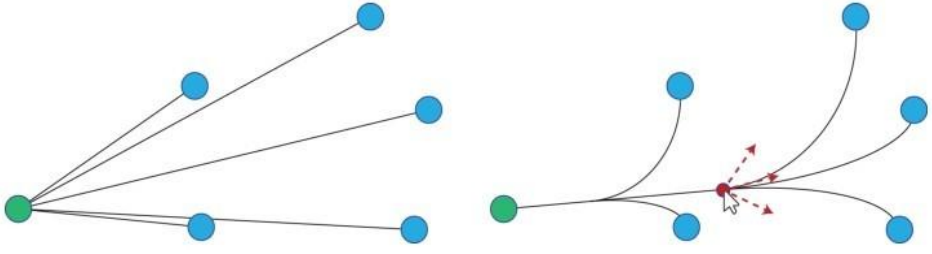
图可视化中的交互

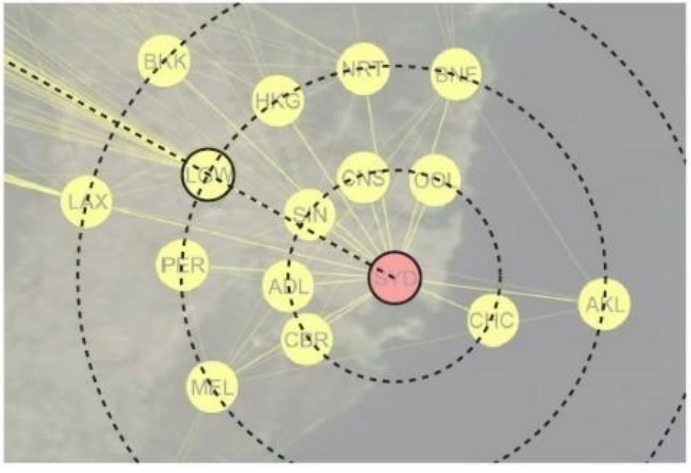
- 图的拓扑简化和边绑定的目的是解决规模较大的图存在的视觉混杂问题。常见的交互方法
  - 基于视点的交互
  - 基于图元的交互
  - 基于图结构的交互

基于视点的交互

- 基于视点的交互是指用交互手段来预测和帮助用户在图中切换视点。视点交互中比较常规的方法包括界面的平移、缩放、旋转等操作
- 近年来，出现了一些跟踪人眼和身体移动轨迹的硬件支持这类交互
- 大规模网络可视化常用的交互操作是Link Sliding和Bring & Go技术

Link Sliding和Bring & Go





基于图元的交互

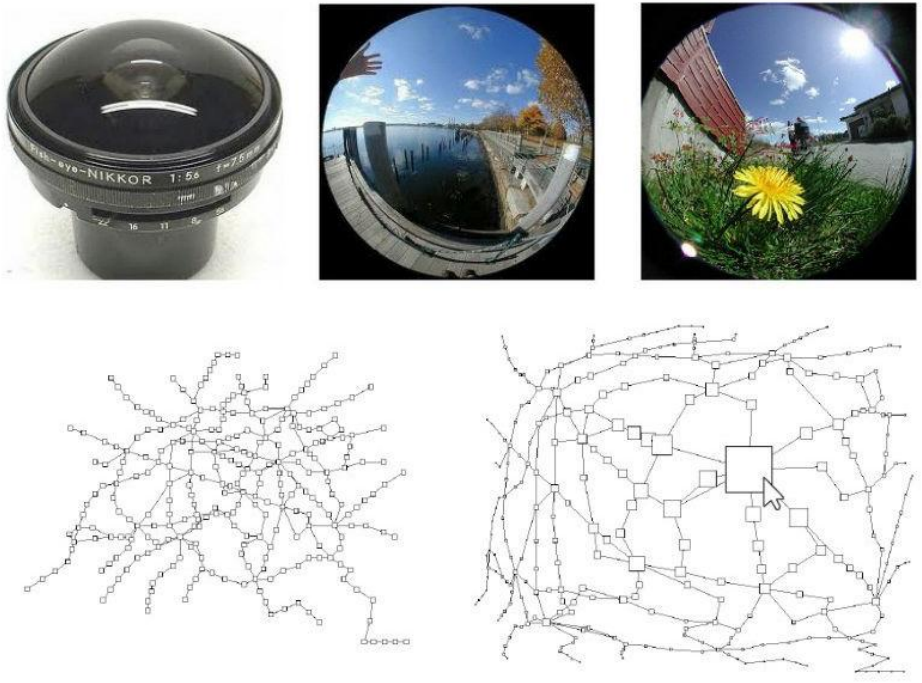
- 基于图元的交互是指对于一个可视化映射元素的交互，如节点的选择、高亮、删除、移动、展开(获取细节)与收缩

- 节点的展开与收缩在大规模网络节点-链接图中应用广泛；节点的收缩可以降低整个布局的视觉复杂度，使布局更加美观；节点的展开配合视点交互可以使用户的注意力聚焦到感兴趣的局部数据

基于图结构的交互

- 基于图结构的交互是一种宏观结构的交互手段，核心思想是“焦点+上下文”(Focus+Context)技术

- 最著名的方法是鱼眼(Fisheye)技术及其变种。鱼眼是一种极端的广角镜头技术，它使用一种焦距极短并且视角接近于180°的镜头，在突出正前方物体的基础上，覆盖视角所及的最大范围



网络数据可视化的挑战

- 网络和层次数据可视化方法所面临的挑战主要来自图的规模
- 用户对数据的认知能力和感知能力也不尽相同，构成了可视化的另一大挑战。外行与数据的领域专家对数据的认知水平不同
- 网络和层次数据可视化面临的挑战也是衡量一个可视化好坏的评价标准

- 一种好的可视化设计往往是用户一部分需求的最大化满足，带有强烈的目的性，它不仅要考虑到数据的特殊性质，还要兼顾用户对数据的认知水平以及用户对数据的可视化需求