



大数据与商务智能

时空数据挖掘

肖升生 博士

xiao.shengsheng@shufe.edu.cn



课程导入：时空数据不断累积

■ 人类活动产生的时空数据

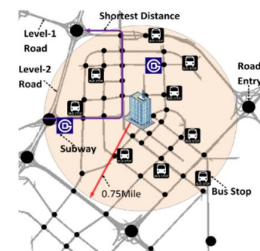
- Active recording

- Travel logs
- Sport analysis
- Check-ins
- ...



- Passive recording

- Credit card transactions
- Public transit records
- Mobile phone signal, Wi-Fi...
-

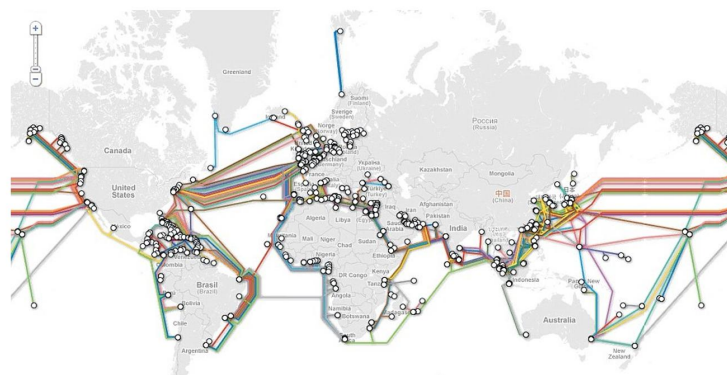
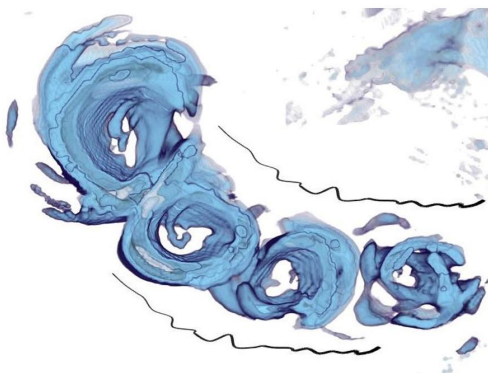




课程导入：时空数据不断累积

■ 非人类活动产生的时空数据

- Mobility of transportation vehicles
 - Taxis, buses, trucks,...
 - Air planes, ferries, cruise,...
- Mobility of Animals
 - Migration: Birds, zebra, tiger
- Mobility of natural phenomena
 - Hurricane, tornado,...





讲授提纲

- 01** 应用场景及挑战
- 02** 时空数据管理
- 03** 时空数据可视化
- 04** 时空数据挖掘算法
- 05** 商业案例分析



讲授提纲

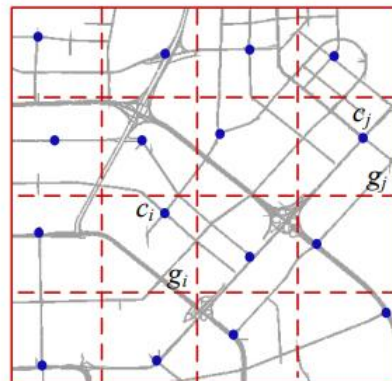
- 01** 应用场景及挑战
- 02** 时空数据管理
- 03** 时空数据可视化
- 04** 时空数据挖掘算法
- 05** 商务案例-餐饮业时空数据挖掘



应用场景：基于位置的服务

■ 地图和导航

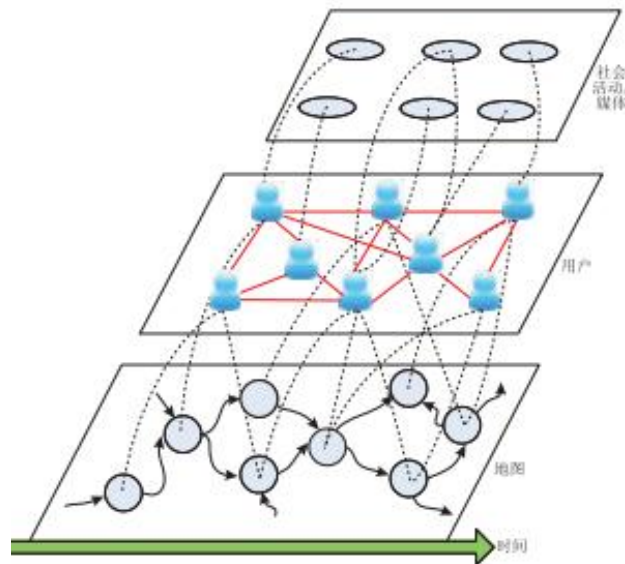
- 智能交通
 - 拼车、路径选择
 - 公共交通需求预测（共享单车等）



■ 基于位置的社交网络 (LBSN)

- POI推荐、社区推荐、朋友推荐、旅行线路推荐等

■ 基于位置的广告 (LBA)



[1] S. Ma, Y. Zheng and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," ICDE, 2013

[2] 刘树栋、孟祥武, 基于位置的社会化网络推荐系统[J], 计算机学报, 2015: 38 (2)



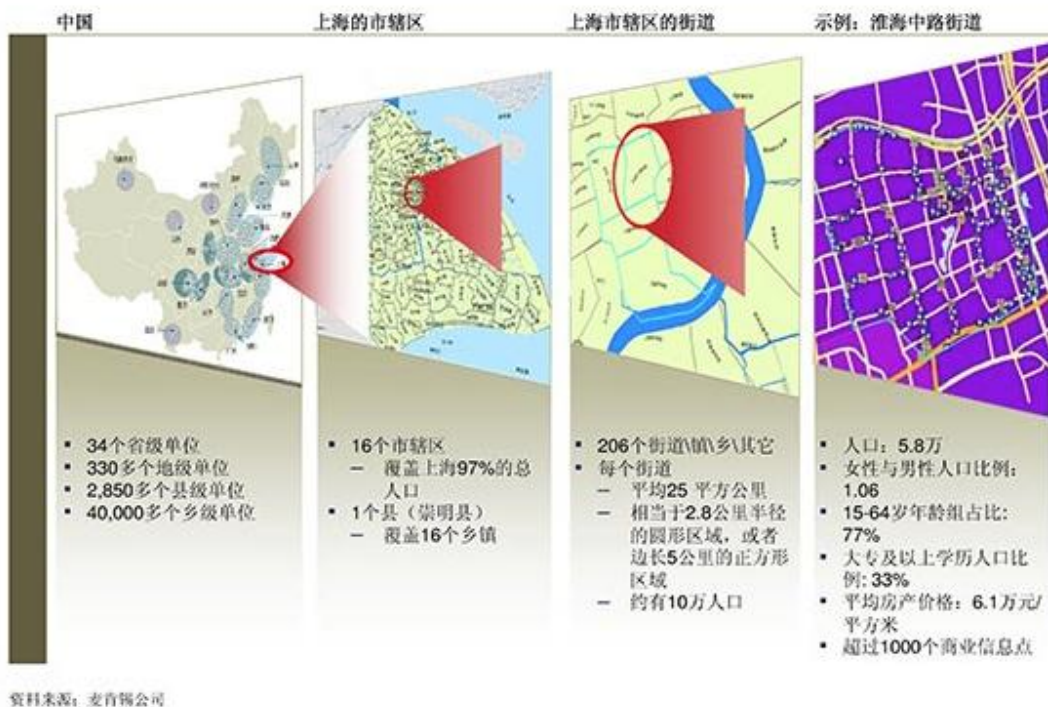
应用场景：地理信息系统

■ 地理信息系统

- 捕获、存储、操作、管理地理数据
- 空间统计等
- 制图学、地理投影

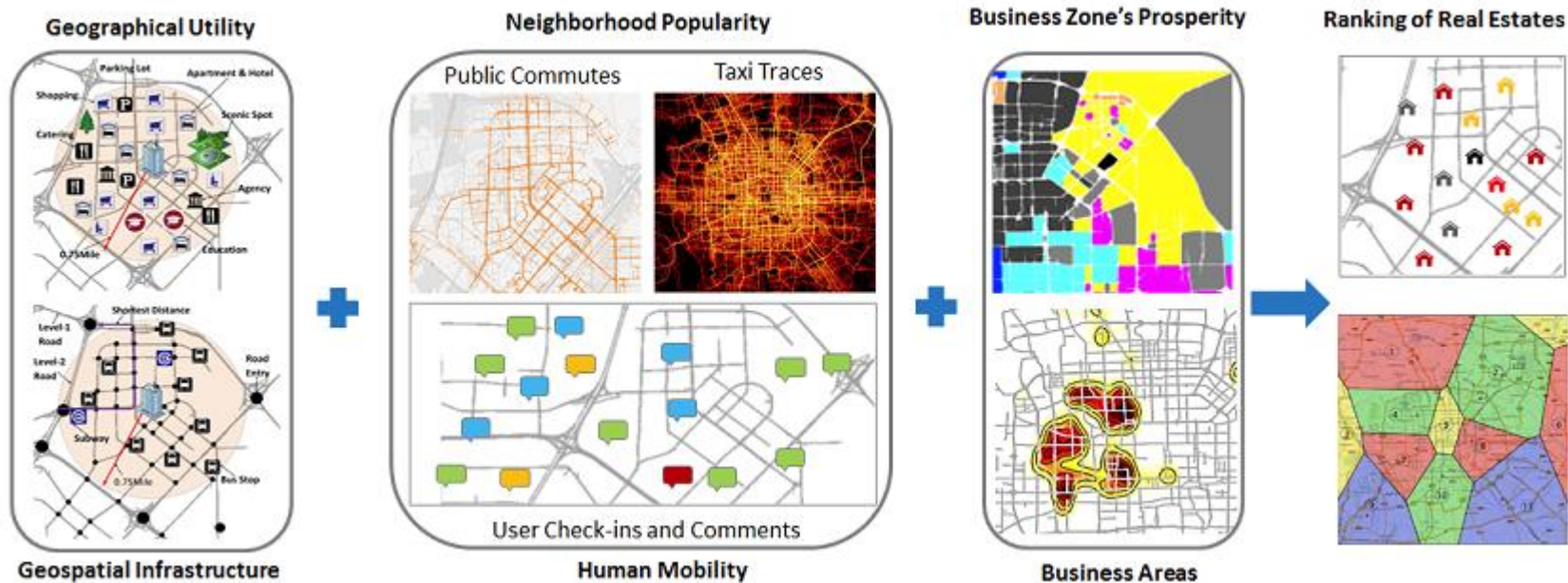
■ 商业地理： Business Geography

商业地理分析使得逐步深入到微观层面的分析成为可能



应用场景：商业选址

■ 店铺选址





应用场景：营销分析

■ Huff模型

- 商店吸引力：满足顾客需求的能力
- 顾客到商店的时间或距离





应用场景：城市交通管理





时空数据挖掘的重要性

■ 在各行各业有显著社会重要性

- 地理气象局
- 生态与环境管理
- 公共安全
- 智能交通
- 流行病管理
- 其他：如地球科学、气候学、物联网等

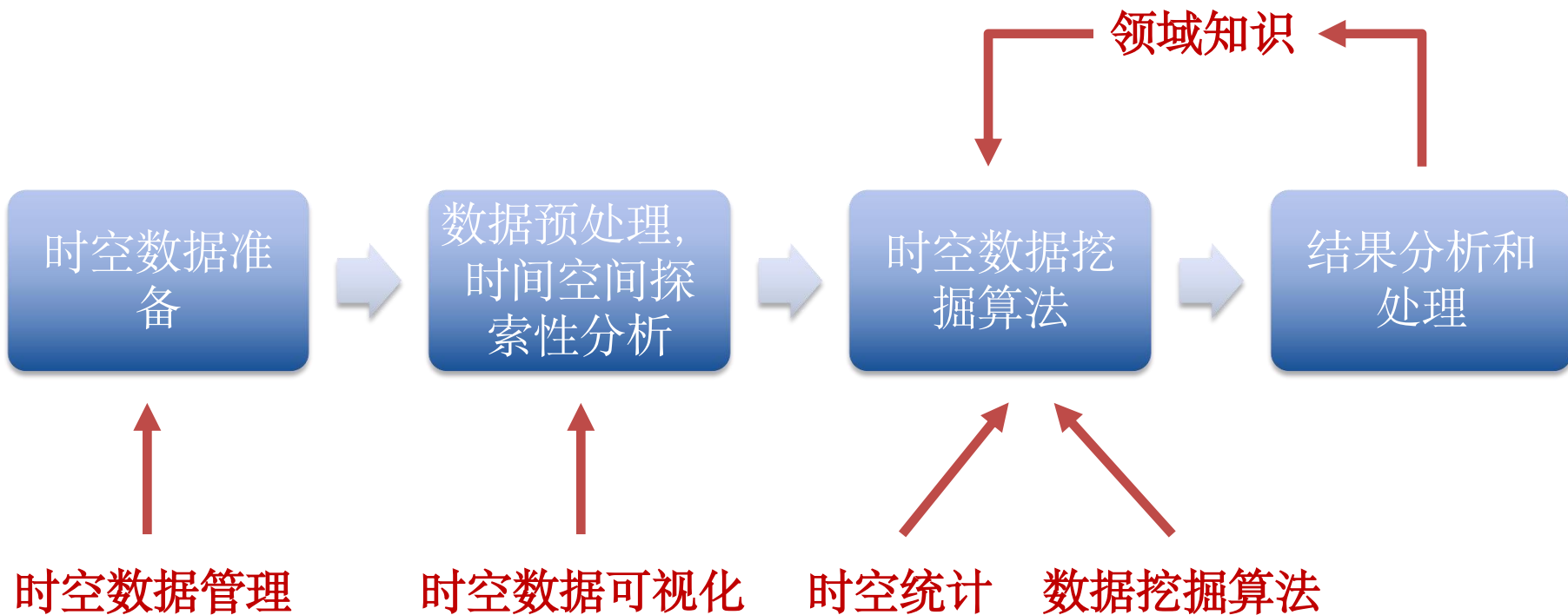


时空数据挖掘的挑战

- 复杂的时空数据类型和关系
- 空间依赖性，独立同分布假设失效
- 嵌入在连续的时间和空间中，非离散
- 空间的异质性和时间的非平稳性
- 多尺度效应

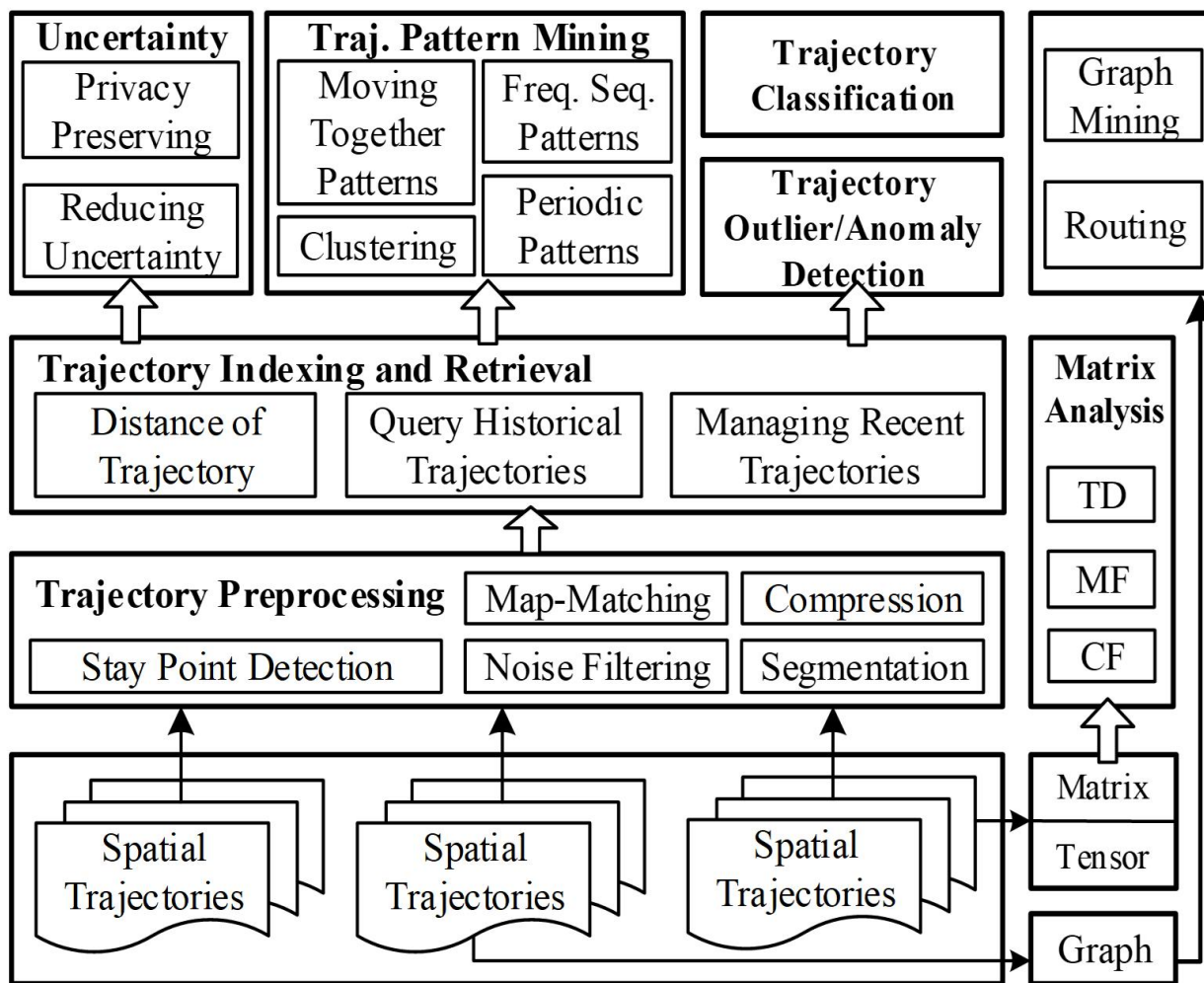


时空数据挖掘的工作流程





时空数据挖掘的框架图





讲授提纲

- 01** 应用场景及挑战
- 02** 时空数据管理
- 03** 时空数据可视化
- 04** 时空数据挖掘算法
- 05** 商务案例-餐饮业时空数据挖掘



时空数据库

- 时空数据库：管理空间、时态以及移动对象数据
 - 多维度、多类型、动态变化、更新快等特点
 - 传统关系型数据库不适应时空数据的处理
- 时空数据库主要类型
 - 空间数据库
 - 处理点、线、区域等二维数据
 - 时态数据库
 - 管理数据的时间属性
 - 移动对象数据库
 - 管理位置随时间连续变化的空间对象
 - 选择查询 (range querying) 和最近邻查询 (KNN querying)



时空数据管理技术

- 时空索引技术
- 轨迹数据管理技术
- 图数据管理技术
- 流数据管理技术

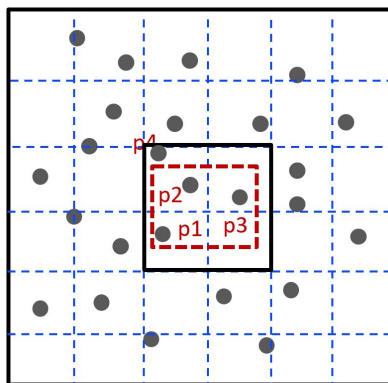
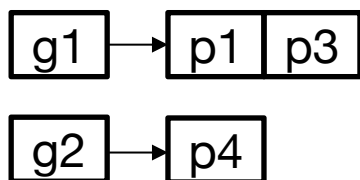


关键技术：时空索引技术

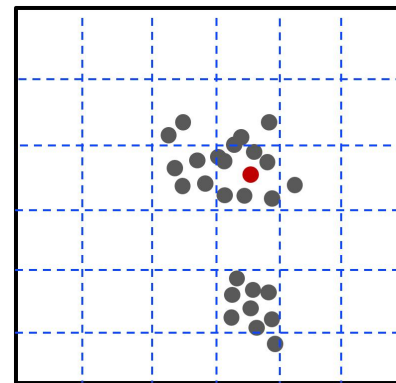
- 有效支持对对象时空范围的查询索引
 - 检索过去
 - 检索当前和未来位置
- 基于空间划分的索引技术
 - **Grid-based**
 - **Quad-tree**
 - **K-D tree**
- 基于数据划分的索引技术
 - **R-Tree**

时空索引技术: Grid-based

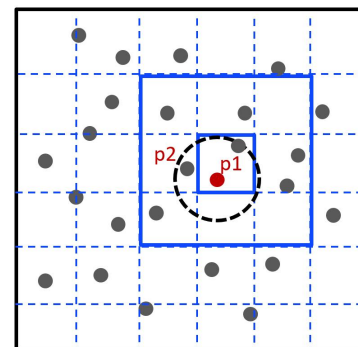
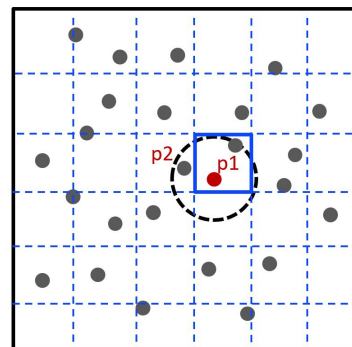
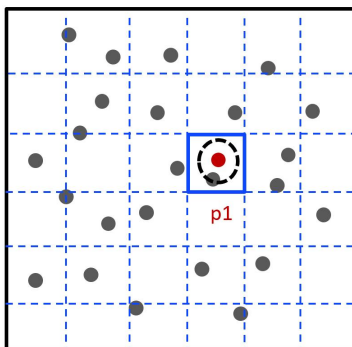
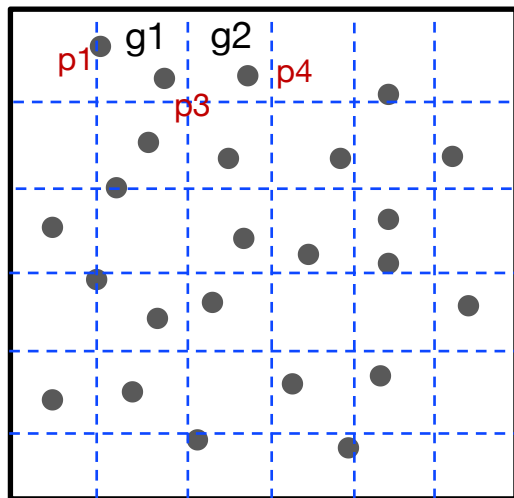
■ Grid-based 技术



区域查询



Unbalanced data

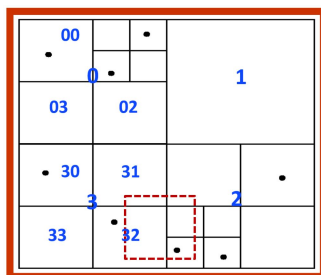
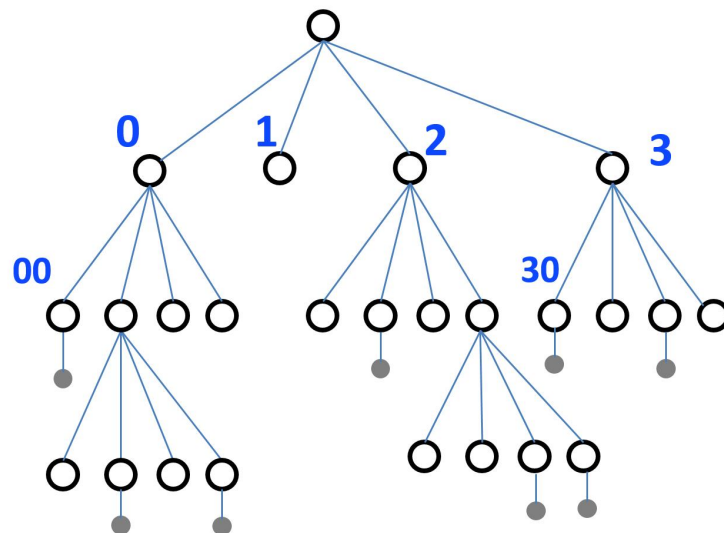
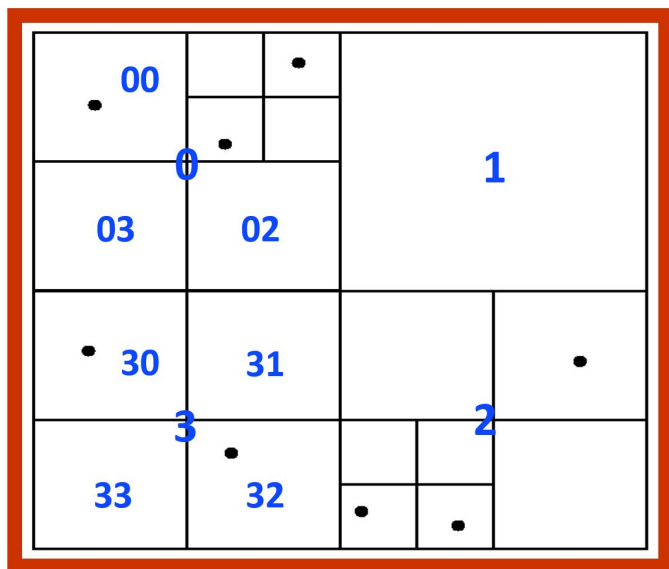


最近邻查询

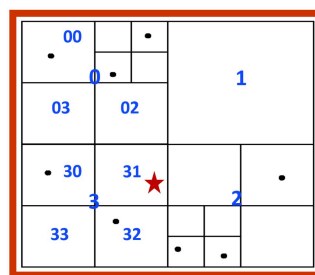
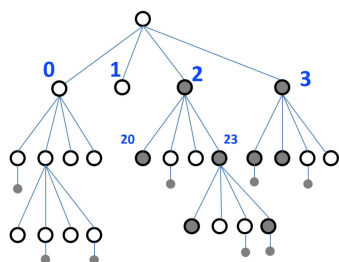


时空索引技术：Quad-Tree

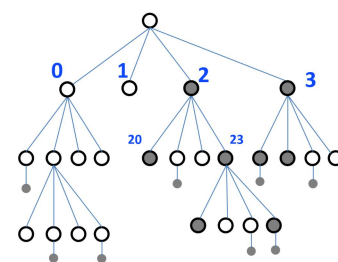
■ Quad-Tree 技术



区域查询



最近邻查询

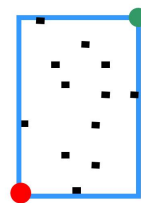
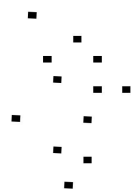




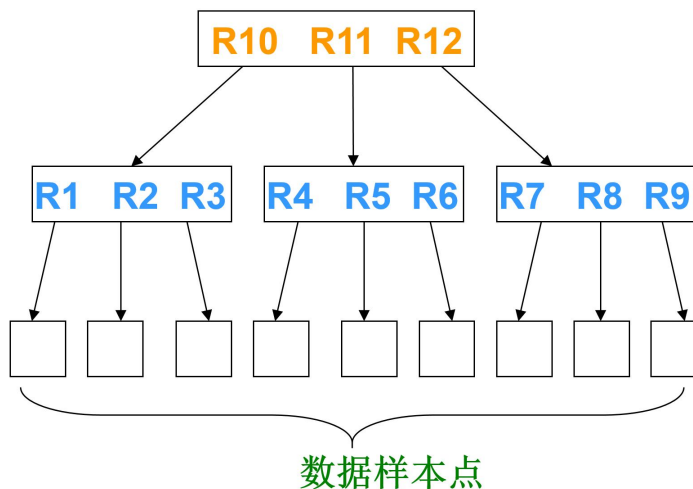
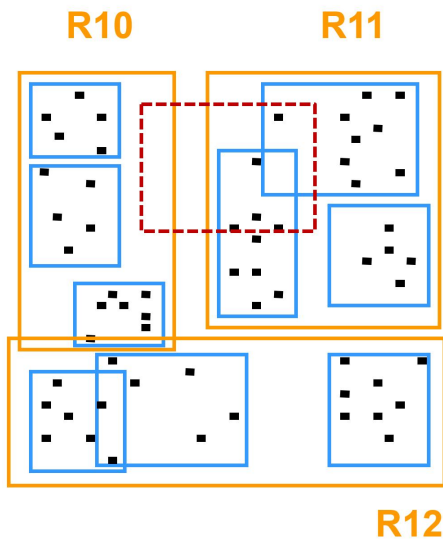
时空索引技术：R-tree

■ R-tree

- 构建 Minimum Bounding Rectangle (MBR)



$$\text{MBR} = \{(\text{L.x}, \text{L.y}) (\text{U.x}, \text{U.y})\}$$



数据样本点

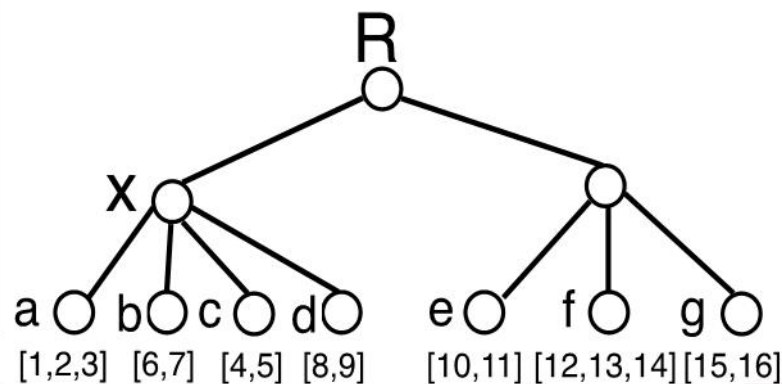
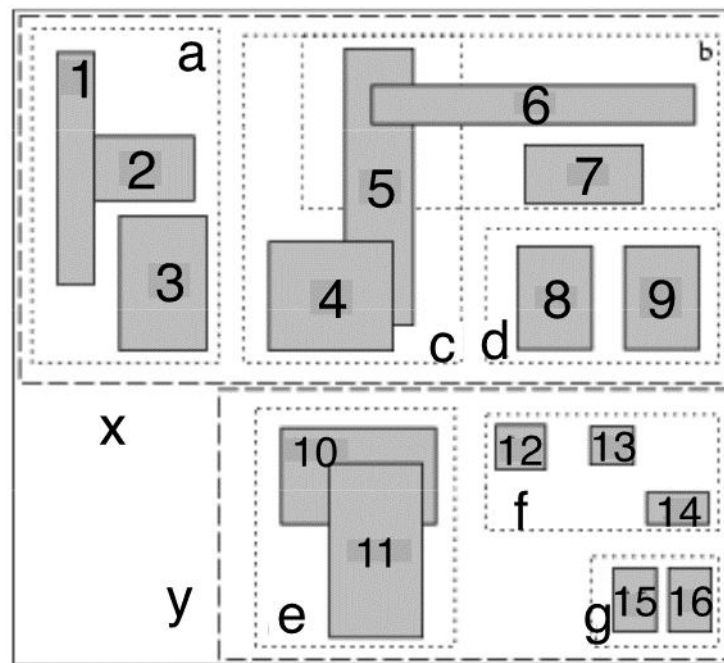


时空索引技术：R-tree

■ R-tree

1. 从根节点开始确定子节点
 2. 递归查询相应的子节点
- 如：查询区域对象5

- 从根节点确定子节点X
- 根据X确定子节点b和c
- 从b没有找到区域对象5
- 从c找到区域对象5



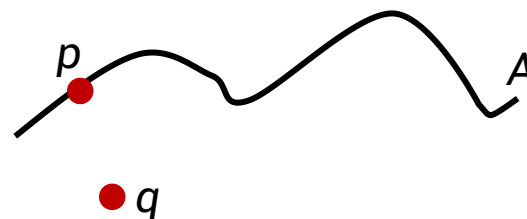


关键技术：轨迹数据管理技术

■ 移动对象数据库

■ 距离公式

- 点 q 与轨迹 A 之间的距离
 - $D(q, A) = \min_{p \in A} D(q, p)$
 - $\text{Sim}(Q, A) = \sum_{q \in Q} e^{-D(q, A)}$: 多个点与轨迹 A 之间的相似度，指数函数将更大的权重给与轨迹 A 距离更相近的点
- 两个轨迹之间的距离
- 两个轨迹段之间的距离





轨迹数据管理：两个轨迹之间的距离

■ 轨迹A与轨迹B之间的距离，多种衡量方式

- Closest-Pair距离

- $CPD(A, B) = \min_{p \in A, p' \in B} D(p, p')$

- Sum-of-Pairs距离

- $SPD(A, B) = \sum_{i=1}^n D(p_i, p'_i)$

- 两段轨迹长度相等

- Dynamic Time wrapping (DTW)距离

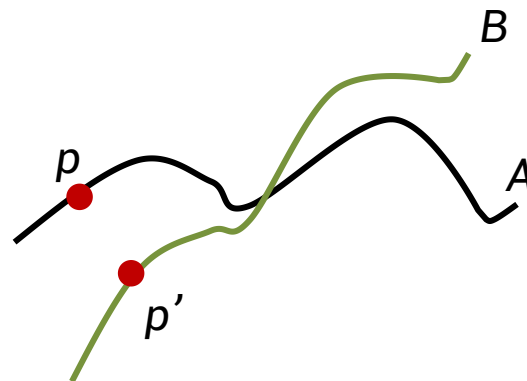
- 允许多次“重复”一些点，以获得最好的对齐，解决轨迹长度相等的限制

- 假设A、B的长度分别为n和m；Head(*)为序列的第一个点，Rest(*)为其余点

$$DTW(A, B) = \begin{cases} 0, & n = 0 \text{ and } m = 0 \\ \infty, & n = 0 \text{ or } m = 0 \\ D(\text{Head}(A), \text{Head}(B)) + \min \begin{cases} DTW(A, \text{Rest}(B)) \\ DTW(\text{Rest}(A), B) \\ DTW(\text{Rest}(A), \text{Rest}(B)) \end{cases} \end{cases}$$

- **缺点：**容易受噪音的影响；不满足三角不等式定理

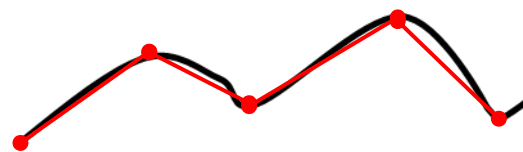
- 其他距离公式：LCSS（允许跳过一些噪音点）、EDR（考虑具有相似公共子序列）、ERP（结合DTW和EDR）等





轨迹数据管理：两个轨迹段之间的距离

■ 两个轨迹段 L_1 和 L_2 之间的距离公式



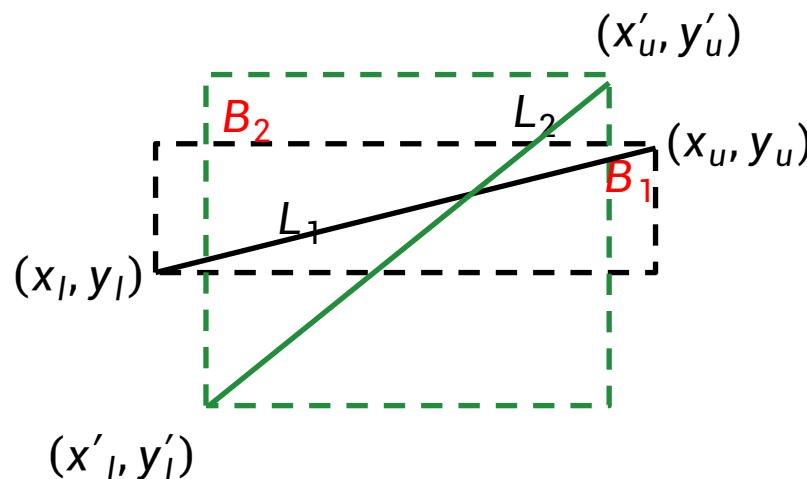
- 基于最小矩阵边框 (minimum bounding rectangle, MBR-based)
- 如图所示，轨迹段 L_1 和 L_2 的MBR分别是 $B_1\{(x_l, y_l), (x_u, y_u)\}$ 和 $B_2\{(x'_l, y'_l), (x'_u, y'_u)\}$

$$D_{min}(B_1, B_2) = \sqrt{(\Delta([x_l, x_u], [x'_l, x'_u]))^2 + (\Delta([y_l, y_u], [y'_l, y'_u]))^2}$$

$$\text{Where } \Delta([x_l, x_u], [x'_l, x'_u]) = \begin{cases} 0 & [x_l, x_u] \cap [x'_l, x'_u] \neq \emptyset \\ x'_l - x_u & x'_l > x_u \\ x_l - x'_u & x_l > x'_u \end{cases}$$

• Trajectory-Hausdorff 距离

- 三个带权重距离的和
 - 两个轨迹段间的垂直距离
 - 平行距离
 - 夹角距离





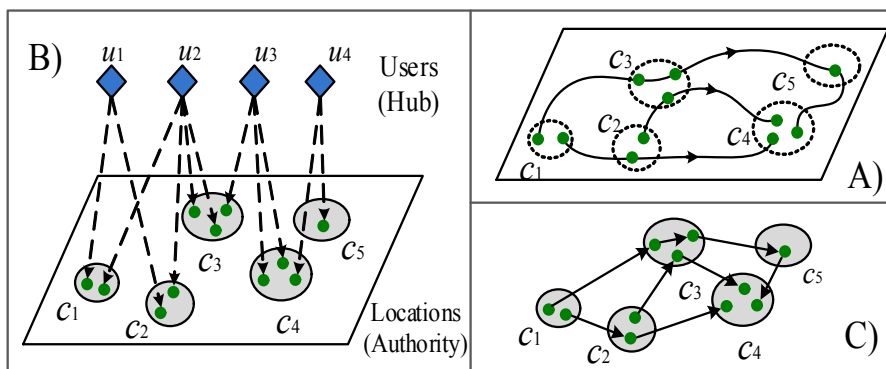
关键技术：图数据管理技术

■ 各种各样的网络

- 不同地区间的人口流动
- 道路上的交通网络
- POI网络等

■ 可使用带有时空属性的图对时空数据进行建模

- 采用基于图的数据挖掘算法，包括目前流行的图神经网络模型等





关键技术：流数据管理技术

■ 传感器感知的数据

- 由于大量的传感器数据都以流的形式输入，高效的流数据库技术是时空数据管理层的基石

■ 三个主要研究问题

- 流数据采样
- 持续性数据查询
- 流数据并行计算



讲授提纲

- 01** 应用场景及挑战
- 02** 时空数据管理
- 03** 时空数据可视化
- 04** 时空数据挖掘算法
- 05** 商务案例-餐饮业时空数据挖掘



时序数据可视化

■ 时序数据

- 随时间变化
- 带有时间性质

■ 例子

- 气温变化
- 股市
- 系统登录
- 历史版本
-



时序数据可视化：基本工具

■ 简单图 (matplotlib)

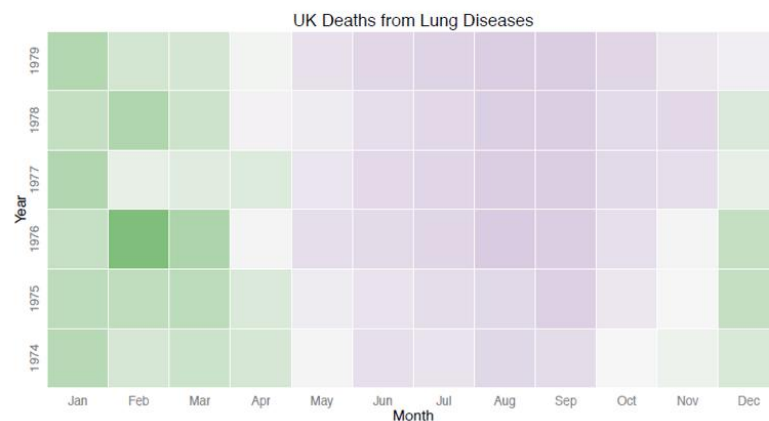
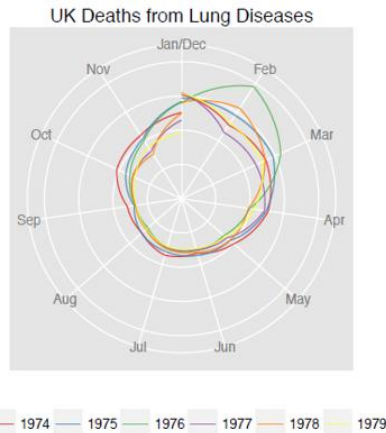
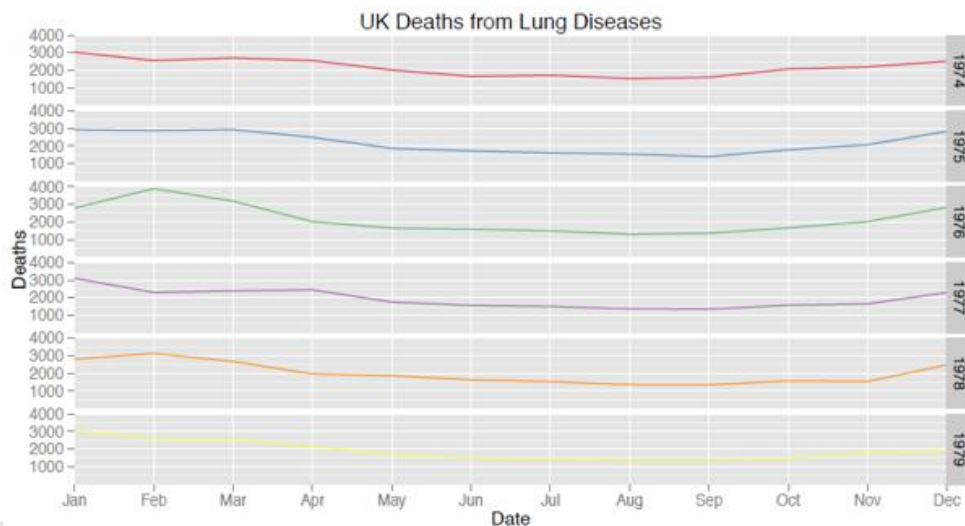
- 折线图 `.pyplot.plot`
- 柱状图 `.pyplot.bar`
- 面积图

■ 复杂图

- 多折线图
- 星形图
- 热图
-

■ 更多资源

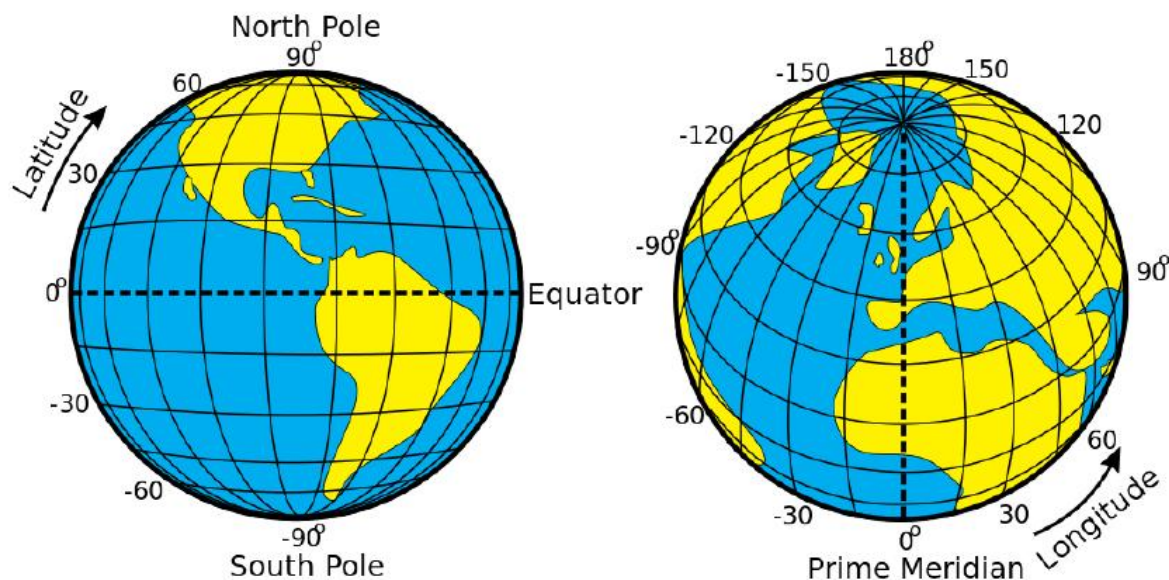
- <http://survey.timeviz.net/>





空间数据可视化

- 空间数据是指带有物理空间坐标的数据
- 地理空间数据
 - 真实的人类生活的空间
 - 经纬度





空间数据可视化：地理信息格式

■ Shapefiles

- 1990S早期GIS软件常使用
- *.shp 包含几何信息
- *.shx 包含索引信息
- *.dbf 包含特征信息

■ GeoJSON

- JSON标准的针对web的地理信息编码方式
- 可以代表一个几何图形、特征以及特征集
- 几何图形包括点、线、多边形、多个点、多条线等
- 可以往JSON格式添加其他数据



空间数据可视化：地理信息格式

■ GeoJSON例子

```
{ "type": "FeatureCollection",  
  "features": [  
    { "type": "Feature",  
      "geometry": { "type": "Point", "coordinates": [102.0, 0.5] },  
      "properties": { "prop0": "value0" }  
    },  
    { "type": "Feature",  
      "geometry": {  
        "type": "LineString",  
        "coordinates": [  
          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]  
        ]  
      },  
      "properties": {  
        "prop0": "value0",  
        "prop1": 0.0  
      }  
    },  
    { "type": "Feature",  
      "geometry": {  
        "type": "Polygon",  
        "coordinates": [  
          [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0],  
            [100.0, 1.0], [100.0, 0.0] ]  
        ]  
      },  
      "properties": {  
        "prop0": "value0",  
        "prop1": { "this": "that" }  
      }  
    }  
  ]  
}
```



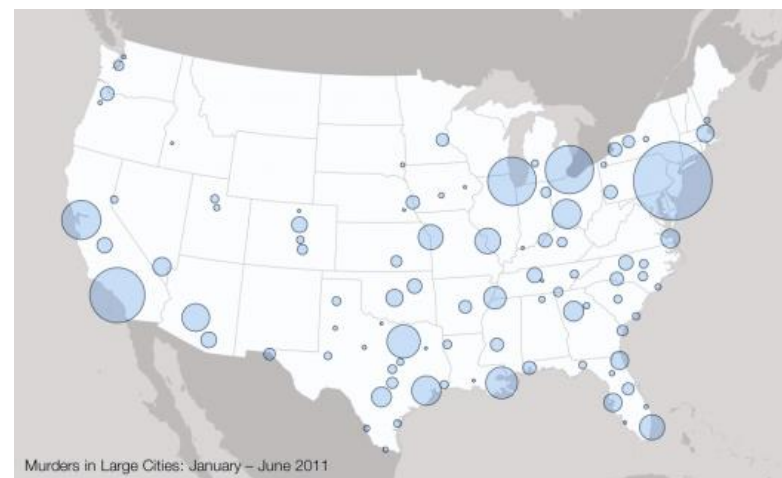
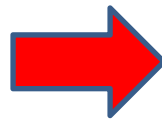
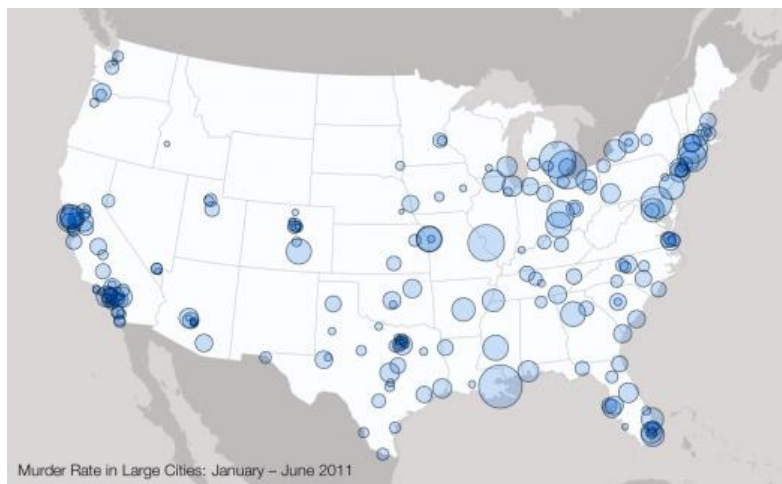
空间数据可视化：地图类型

- 点数据的可视化：符号地图
- 区域数据的可视化
 - Choropleth
 - Cartogram
- 线数据的可视化
 - 流型图
 - 道路图



空间数据可视化：符号地图

- 点数据的可视化
- 将对象根据它的坐标以一定的符号大小及颜色标识到地图上
 - 符号必须直观且符合常识
 - 不同符号数量不宜太多





空间数据可视化

■ 时间+地图





空间数据可视化：线数据可视化

- 线数据通常指连接两个或更多地点的线段或者路径
 - 长度属性
 - 连接关系



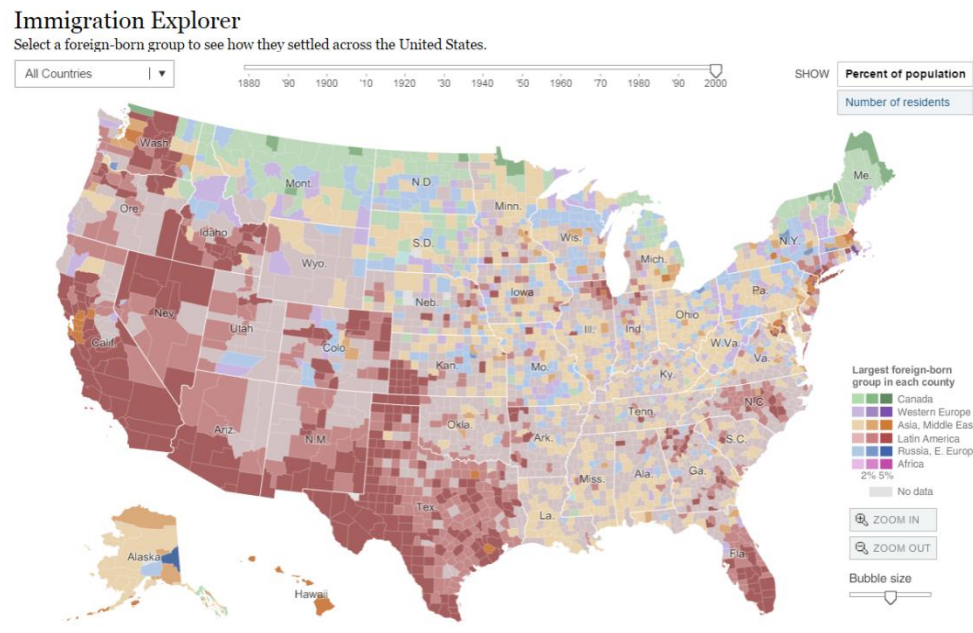
Facebook全球用户之间的好友关系



空间数据可视化：区域数据的可视化

■ Choropleth 地图

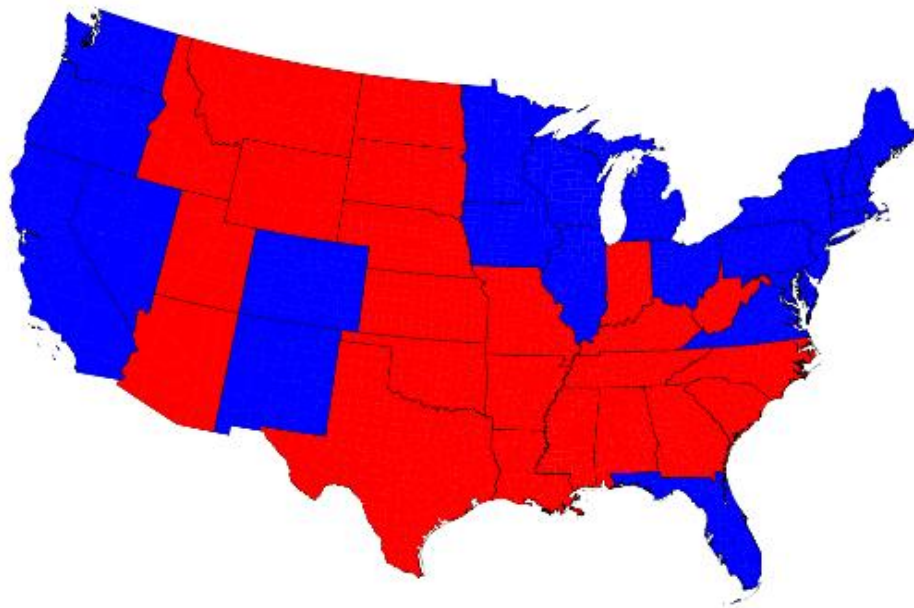
- 假设数据的属性在一个区域内部平均分布
- 一个区域用同一种颜色来表示其属性
- 数据最好标准化（百分比、比例等）
- 用**颜色**及其强度来显示数据的内在模式
- 必须合理地使用颜色
 - 多种颜色，无强度
 - 多种颜色，有强度
 - 单个颜色，有强度





空间数据可视化：区域数据的可视化

- Choropleth 地图的误导
- 数据分布和地理区域大小的不对称
 - 大量数据集中于人口密集的区域，而人口稀疏的地区却占有了大多数的屏幕空间，用大量屏幕空间来表示小部分数据的做法对空间的利用非常不经济
 - 这种不对称还常常会造成用户对数据理解的错误

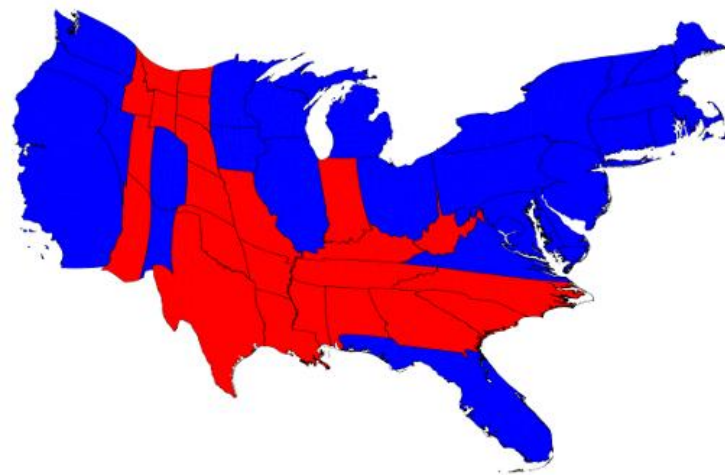


08年美国总统选举



空间数据可视化：区域数据的可视化

- 保证用户的思维连贯性
- 相似的形状，被描绘面积的邻接关系
- Cartogram地图：按照地理区域的属性值对各区域进行适当变形，克服Choropleth地图对空间展示的不合理性
 - Area Cartogram
 - 按照区域大小对地图处理
 - 保证区域之间的连接和相对位置不变
 - 区域仍保持连续
 - 非连续性Cartogram
 - 按照区域大小对地图进行处理
 - 保持区域的原始形状
 - Dorling Cartogram
 - 不再保持区域的形状和连接关系
 - 用其他相比例的形状替代



地图上的每个州根据人口做缩放



空间数据可视化：工具

■ Python packages

- Matplotlib, Seaborn, 各种统计图
- 空间数据: folium
 - <https://python-visualization.github.io/folium/quickstart.html>
- 其他: pyecharts, plotly, bokeh, geopandas等



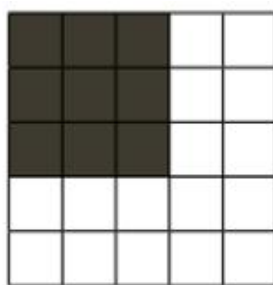
讲授提纲

- 01** 应用场景及挑战
- 02** 时空数据管理
- 03** 时空数据可视化
- 04** 时空数据挖掘算法
- 05** 商务案例-餐饮业时空数据挖掘



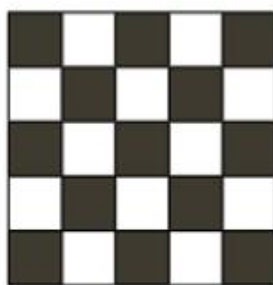
时空数据统计

- 时空数据具有空间自相关性、异质性、时间自相关性等特性，时空统计有别于经典统计
- 不满足独立同分布的假设
 - [Tobler's first law of geography](#): 地理学第一定理
 - 所有事物都与其他事物相关，此联系随着距离增加而递减



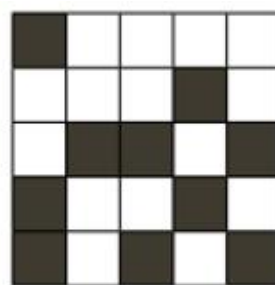
(a)

空间正相关



(b)

空间负相关



(c)

空间无相关性

- 空间异质性: 属性值的空间差异性 (地理学第二定理, [Michael Goodchild](#))
 - 全局模型与局部模型不一致
 - 某些时候可考虑对数据从空间上进行切片 (分组), 再进行DM



传统DM算法在时空数据中的应用

- 将时间和空间属性当作常规的数据特征，使用经典的数据挖掘技术，将导致信息的丢失
 - 空间连续性（空间划分方式会影响结果）
 - 自相关、异质性， etc.
 - 噪音的问题
- 时空数据挖掘算法
 - 空间自回归
 - 地理加权算法
 - 空间聚类
 - 空间决策树
 - 空间关联规则分析



空间自回归

- 经典的回归模型无法将空间的相关性考虑进去

- Spatial Autoregression (SAR)

- $y = \rho W y + X\beta + \epsilon$
- $y \in R^n, X \in R^{n \times k}, \beta \in R^k, \epsilon$ 是观测误差，服从正态分布
- $\rho W y$: 空间自相关项，用于对因变量 y 的元素之间的空间依赖强度进行建模， $\rho \in [0, 1)$ 空间自相关参数， $W \in R^{n \times n}$ 矩阵解释空间数据之间的空间关系

- $$W = \begin{pmatrix} 0 & w_{12} & \dots & w_{1n} \\ w_{21} & 0 & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & 0 \end{pmatrix}$$

- w_{ij} 是指 i 和 j 的空间相关性，即相似性
 - 基于距离的 (K-function)
 - Moran's I measure (MI) $\in [-1, 1]$ (基于邻居的)
 - ...

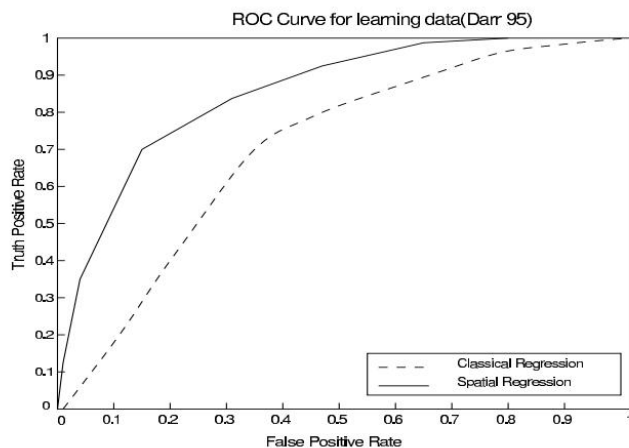
- 考虑被附近的自变量影响的情况

- $y = \alpha + \beta x + \delta W X$

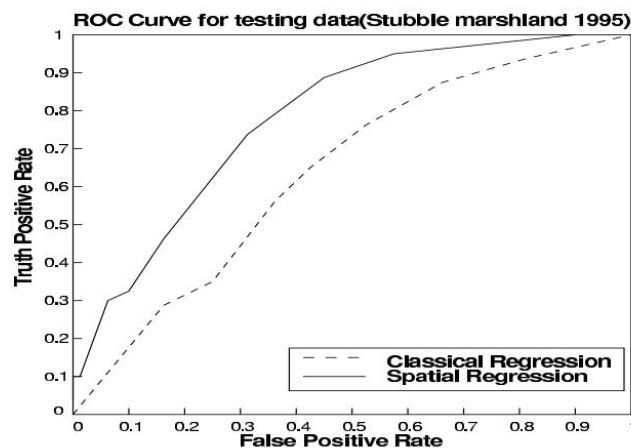


空间自回归

■ 空间溢出效应



(a) ROC curves for learning



(b) ROC curves for testing

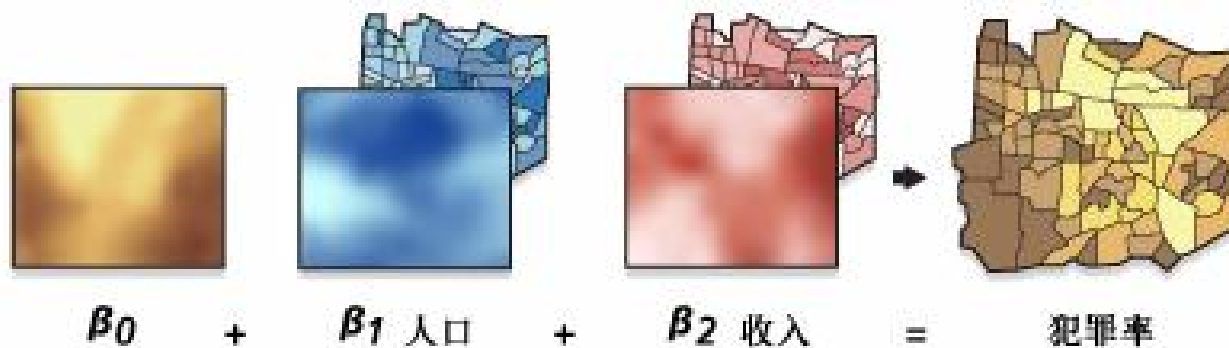
■ 时间自回归

- 时序数据的自相关性
- 各种不同模型: AR, ARIMA, etc.



地理加权回归

- 考虑空间异质性
- Geographically Weighted Regression (GWR)
 - 局部回归模型
 - $y = X\beta' + \epsilon'$
 - β' 与 ϵ' 地理位置相关





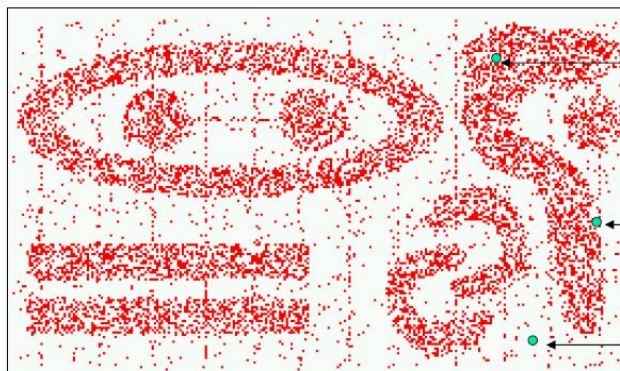
空间聚类

■ CLARANS (Ng and Han, 1994)

■ DBSCAN

- Density-Based **Spatial** Clustering of Applications with Noise
- 算法的思想是寻找具有足够的高密度的连通区域，而低密度区域的点则作为孤立点
- 一个点的密度可以看作所有样本点与此点的相似度之和
- 参数：密度水平和近邻参数 (ϵ , $MinPts$) - 难以选择
- 优点：可以发现任意形状类

OPTICS算法

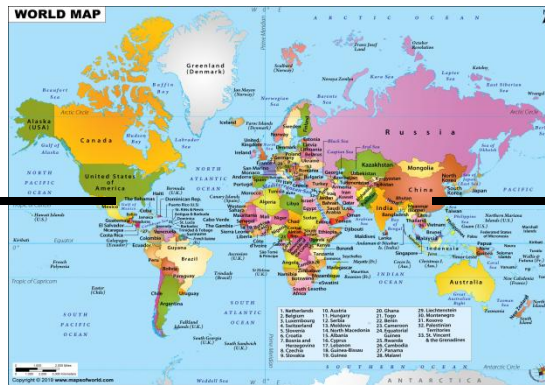


核心对象

非核心对象(边界点)

噪声样本

空间关联规则分析



- 空间关联规则分析：事件在时间和空间上同时发生 (Spatial Colocation)

	关联规则	Co-location 规则
潜在空间 (underlying spaces)	离散集	连续空间
物品类型	物品	事件 / 布尔空间特征集
集合	交易 (T)	邻居 (N)
衡量标准	support	participation index
条件概率指标	$\text{Pr.}[A \text{ in } T \mid B \text{ in } T]$	$\text{Pr.}[A \text{ in } N(L) \mid B \text{ at location } L]$



讲授提纲

- 01** 应用场景及挑战
- 02** 时空数据管理
- 03** 时空数据可视化
- 04** 时空数据挖掘算法
- 05** 商务案例-餐饮业时空数据挖掘



在线用户评论分析

- 整理自Yelp官方公开的商户、点评和用户数据
 - <https://www.yelp.com/dataset>
- 整理好的Yelp数据集包含位于多伦多的所有餐馆信息(biz_res.txt)、所有餐馆截至2017年7月的评论数据(review_res.txt)及所有相关用户个人信息(user_res.txt)。
- 各文件的数据字段名称详见ReadMe.txt，数据字段含义详见Yelp Dataset JSON.pdf