



北京航空航天大学 博士论文答辩  
BEIHANG UNIVERSITY

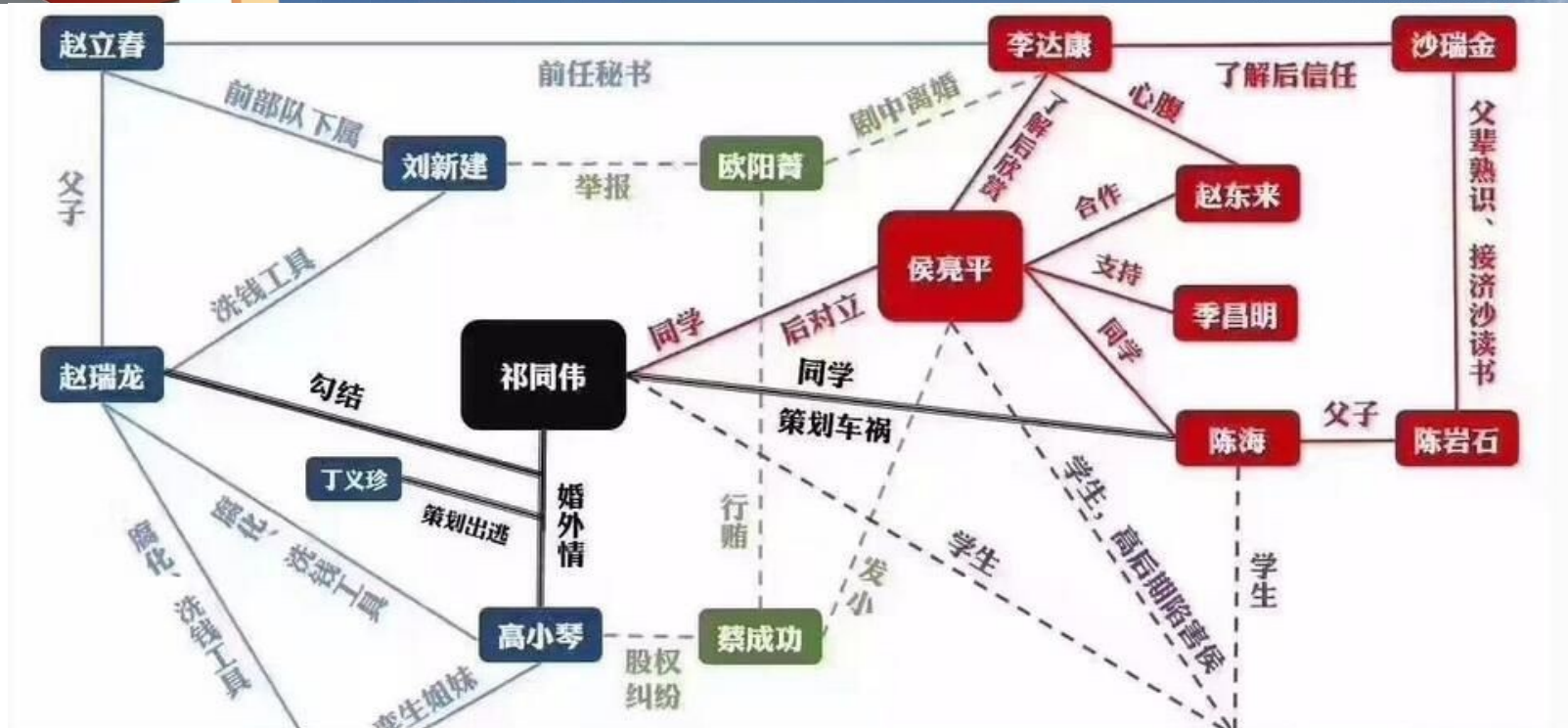
# 复杂网络应用 网络可视化

李春芳

中国传媒大学 计算机学院

2019.4.24





# D3 文件操作

d3.json() : Json 文件

d3.csv() : csv 表格

d3.text() : text 文本

.....

绝大多数浏览器不支持读取本地文件，因此要讲文件放到 Tomcat 服务器下

```
d3.json("a.json",function(error,data){  
    funA(data);  
});
```



# Json 格式文件

**JSON** : **JavaScript 对象表示法** ( JavaScript Object Notation ) 。

JSON 是存储和交换文本信息的语法。类似 XML 。

JSON 比 XML 更小、更快，更易解析。

## 两种调用方式

- 转换为 Json 格式的文本文件
- 拼写 Json 格式字符串

# JSON

- JSON 是纯文本
- JSON 具有“自我描述性”（人类可读）
- JSON 具有层级结构（值中存在值）
- JSON 可通过 JavaScript 进行解析
- JSON 数据可使用 AJAX 进行传输

# JSON 数据格式

- 名称 / 值对

```
{ "employees":  
    [  
        { "firstName":"John" , "lastName":"Doe" },  
        { "firstName":"Anna" , "lastName":"Smith" },  
        { "firstName":"Peter" , "lastName":"Jones" }  
    ]  
}
```

# Json 在线解析工具

<http://www.json.cn/>

```
{
  "nodes":
  [
    {"name": "@", "group": 0},
    {"name": "a", "group": 1},
    {"name": "b", "group": 2}
  ],
  "links":
  [
    {"source": 0, "target": 1, "value": 1},
    {"source": 1, "target": 2, "value": 1},
    {"source": 2, "target": 0, "value": 1}
  ]
}
```



# Json 示例

```
<body>
<h2> 在 JavaScript 中创建 JSON 对象
</h2>

<p>
Name: <span id="jname">
</span><br />
Age: <span id="jage">
</span><br />
Address: <span id="jstreet">
</span><br />
Phone: <span id="jphone">
</span><br />
</p>

<script type="text/javascript">
var JSONObject= {
  "name":"Bill Gates",
  "street":"Fifth Avenue New York 666",
  "age":56,
  "phone":"555 1234567"};
document.getElementById("jname").innerHTML=JSONObject.
name
document.getElementById("jage").innerHTML=JSONObject.ag
e
document.getElementById("jstreet").innerHTML=JSONObject.
street
document.getElementById("jphone").innerHTML=JSONObject.
phone
</script>

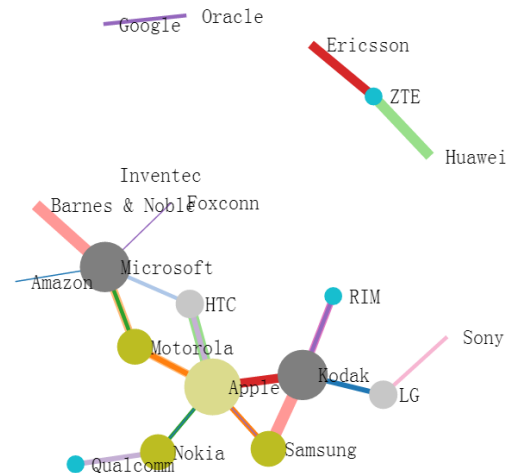
</body>
</html>
```

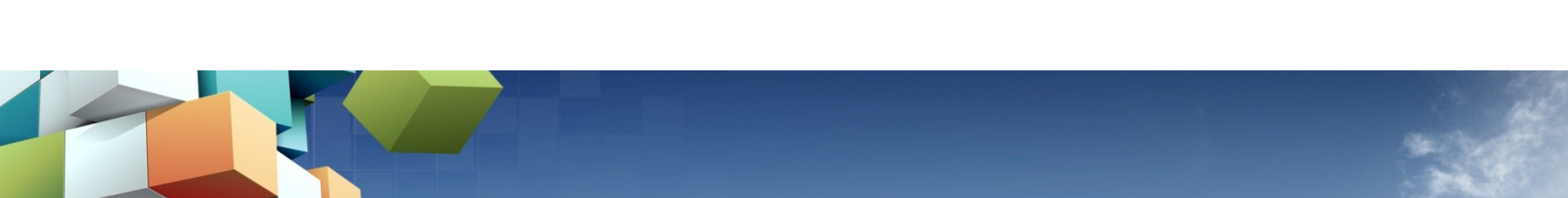





# D3 力导向图的 Json

```
var links = [  
  {source: "Microsoft", target: "Amazon", type: "licensing", weight:1,color:1},  
  {source: "Microsoft", target: "HTC", type: "licensing", weight:3,color:4},  
  {source: "Samsung", target: "Apple", type: "suit", weight:4,color:6},  
  .....];
```





```
var force = d3.layout.force()  
                .charge(-120)  
                .linkDistance(50)  
                .size([width, height]);
```



# 读取 Json 数据绘制力导向图

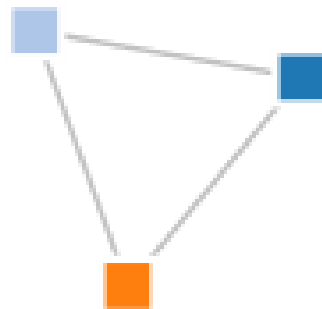
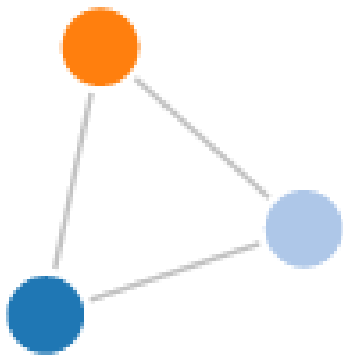
```
d3.json("aa.json", function(error, graph) {  
    force  
        .nodes(graph.nodes)  
        .links(graph.links)  
        .start();  
  
    var link = svg.selectAll(".link")  
        .data(graph.links)  
        .enter().append("line")  
        .attr("class", "link")  
        .style("stroke-width", function(d) { return d.value; });  
  
    var node = svg.selectAll(".node")  
        .data(graph.nodes)  
        .enter().append("circle")  
        .attr("class", "node")  
        .attr("r", 16)  
        .style("fill", function(d) { return color(d.group); })  
        .call(force.drag);  
  
    node.append("title")  
        .text(function(d) { return d.name; });  
  
    force.on("tick", function() {  
        function(d) { return d.source.x; }  
        .attr("y1", function(d) { return d.source.y; })  
        .attr("x2", function(d) { return d.target.x; })  
        .attr("y2", function(d) { return d.target.y; });  
        node.attr("cx", function(d) { return d.x; })  
        .attr("cy", function(d) { return d.y; });  
    });  
    link.attr("x1",
```



# 力导向图要点

- Json 文件格式
- `svg.layout.force` 数据准备：节点、边
- 节点：circle
- 边：line

# 力导向图？





# 小结

数据：节点和边表  
示图或网络

`d3.layout.force()`  
计算节点位置

绘制节点  
圆形：circle  
矩形：rect

绘制边  
无向：line  
有向：path



# 关于 D3 是什么

- Data-Driven Documents
- 数据 ( Data ) 驱动的文档 ( DOM 元素 )
- JavaScript 语言对数据可视化的封装
- 基本绘图元素：SVG 的元素
- 布局：对数据的准备，画图由基本元素完成

套路很深.....

感谢

