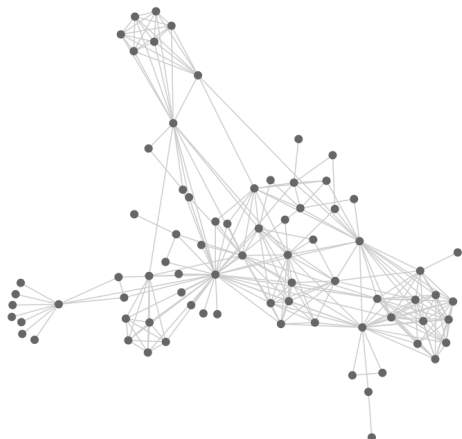



山东大学 计算机科学与技术 学院

可视化技术 课程实验报告

学号：201900130176	姓名：李伟国	班级：智能
实验题目：理解 Force-directed layout		
实验学时：	实验日期：2021/10/25	
实验目的： 用 D3 跑一下 force-directed layout		
硬件环境： 处理器：AMD Ryzen 5 3600 6-Core Processor 3.60 GHz Ram 16.0 GB		
软件环境：		
实验步骤与内容： Force-directed layout： 需要计算一组点的集合中任意成对的点的之间的相互作用力，这经常被叫做 N-body 问题。在 network visualization 中，我们要使用类似与在星球之间计算天体之间的引力，去计算布局。在 network 中的每个节点扮演者相互排斥的带电粒子的角色，而链接连个节点的边扮演着将两个节点拉在一起的弹簧。这这就是 force-directed layout。这里的 force 可以看做是所有点之间两两节点的斥力和边对两两节点的拉力。 可以通过官网的例子来感受一下这种 force-directed 的布局。		
<div><div>Force Strength <input type="range"/> -30.00</div></div> <div><div>Force Strength <input type="range"/> -10.00</div></div>		

通过改变 Force strength，可以发现，force strength 越大，节点彼此之间的距离就会变得越来越小，layout 就会显得越紧凑。

计算 N-body force 最直接的方法就是考虑所有节点彼此之间的相互作用力，并且将每个节点的相互作用力的贡献都加起来。但是这种最初的办法有 N^2 的复杂度，这在实际情况中是很不利的，所以要改进。下面是改进算法。

Improvement: Barnes-Hut Approximation

核心思想是：通过用它们的质心替换一组远距离点来近似远程力。作为少量错误的交换，该方案显著加快了计算速度，复杂度为 $n \log n$ 而不是 n^2 。

该近似方法的中心是 spatial index（空间索引），一个空间的 map 可以帮助我们在一组节点建模成一个质点。在 2 维平面上，我们可以使用 quadtree 这个数据结构，这个数据结构可以递归的将空间中的平面区域细分成 4 个等大的象限。如果实在 3 维的空间中，我们可以类比的使用 octree 将一个立方体分成空间中的 8 个子立方体。

主要有 3 步：

1. 构建空间 索引（spatial index）（quadtree 或者是 octree）
2. 计算质心
3. 估计力（force）

Step 1: construct the Quadtree

以二维的数据点为例。当我们将第一个点插入到 Quadtree，这个被添加到树的最高级别的树根中。

接下来去插入另一个节点，需要将空间进行一次细分来扩展这颗树。在每次后续插入时，可能会添加更细粒度的单元格，直到所有点都位于它们自己的单元格中。

通过网站给的交互式接口来体验一下这个过程。

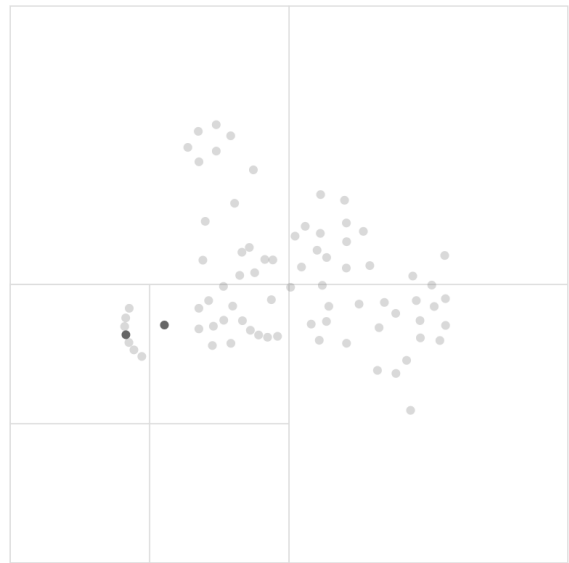
刚开始，原始的输入图像（一些 2 维的输入点）



添加一个节点到 quadtree 中



添加第二个节点到 quadtree 中



此时可以发现，当添加第一个节点后，给节点就位于树的根部(外部最大的矩形框)，当再次添加第二个节点的时候，对空间进行细分，直到该节点单独的位于一个 cell 之中。

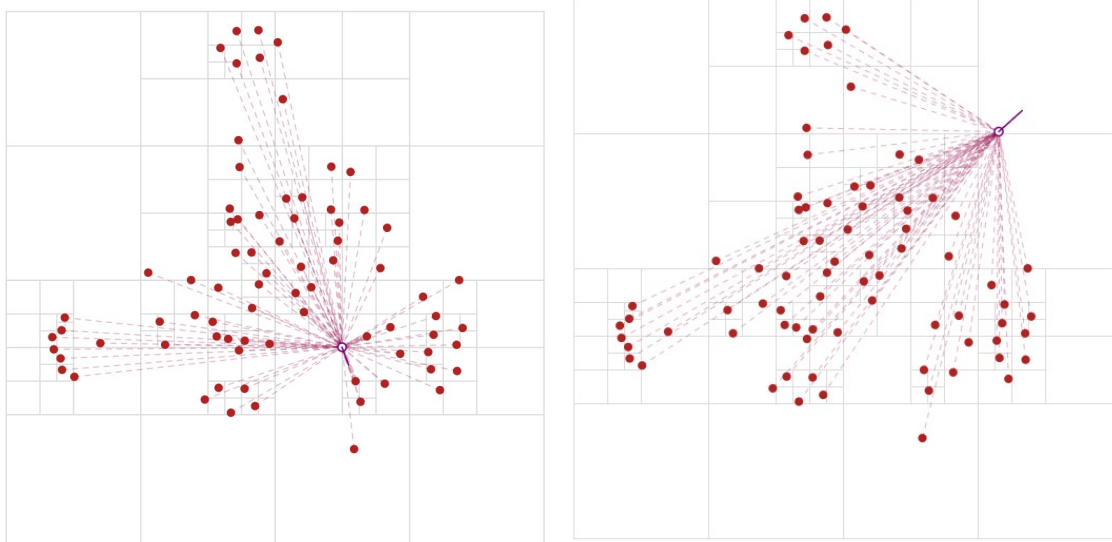
Step 2: calculate Center of Mass

构建好 quadtree 后，就可以对树的每个 cell 计算质心了。计算的方法很简单，每一个树的 cell 的质心就是其四个孩子的中心的加权平均。首先访问 quadtree 的叶子 cell，然后再访问其后续的父亲 cell。合并那些我们向根节点传递个数据。一旦访问（遍历）全部节点，树的每一个 cell 都已经使用位置和它的质心更新过了。

Step 3: Estimate N-Body Forces

当计算给定点所受的力的时候，通过交互式的 web 接口，我们可以很直观的看清楚背会的计算原理。

如图，白色点是给定的点，要计算它和所有点之间的斥力的合力。（这个给定的点类似于高中物理所学的试探电荷，不妨在这里将其称作试探 point）



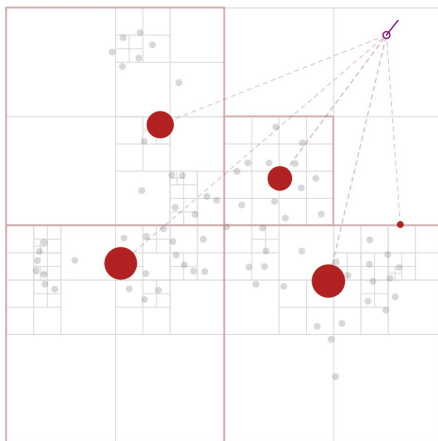
通过移动给定点，可以发现，在不同的位置，其所受的其它节点对其斥力的和力是不一样的。

最开始的方法是：先忽略这个 quadtree 这个结构，通过将所有的节点对给定的点所造成的斥力用向量加法计算其合力。

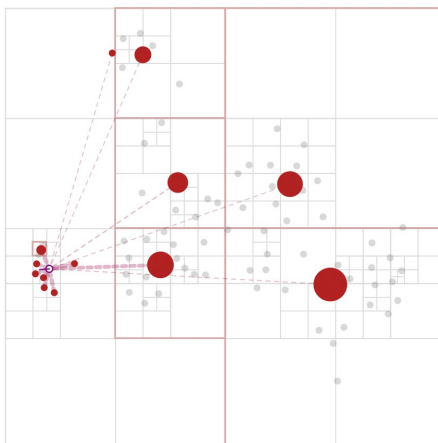
但是我们会使用 quadtree 去代替这个最原始的方法去加速计算和近似的远程的力。不是计算每个节点之间的相互作用力，我们可以计算给定点和质心的作用力就可以了，使用小的 quadtree 的 cell 去计算邻近的点，用大的 cell 去计算远的点的相互作用力。

但是有一个至关重要的点，那就是什么是“long-range”，什么是“short-range”的力？我们既要考虑到树的 cell 的距离，又要考虑 cell 的边长。如果比值 $\text{width}/\text{distance}$ 小于我们设定的 threshold(阈值)，我们就认为是“long-range” forces 并且使用该 cell 的质心。反之，我们将会递归的访问每个子 cell。

当 threshold 设置为 1 的时候，quadtree 树的 cell 的质心会被使用。并且它的内部节点会被忽略，如果样本点的距离到 cell 的中心的距离大于或者等于 cell 的边长（即小于一定的阈值）。

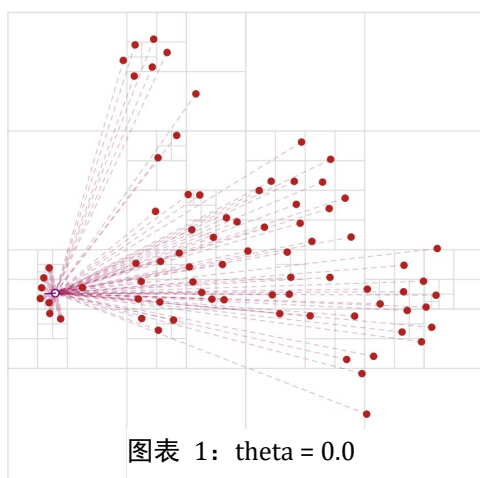


此时样本点到 cell 的中心的距离要大于 cell 的边长，此时就将整个 cell 作为整体来表示对 sample 点的影响



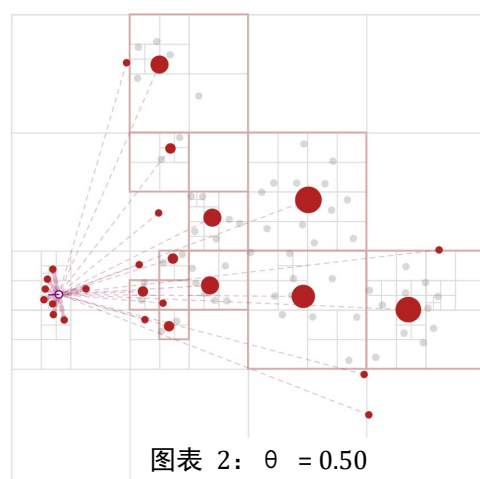
当我们移动了 prob 的位置，可以发现，在离它最近的位置的点，prob 到其 cell 的距离小于 cell 的边长，这是就递归的访问所有的子 cell，而在 prob 的右边的 cell 中，prob 到 cell 中心的距离大于 cell 的边长，此时就将整个 Cell 作为整体访问（很大的红色的点）。

但我们不断的调整 θ 的值, 我们就会发现越来越多的离根节点越近的 CELL 的中心到 sample 点的句距离的比值会轻易的大于该阈值, 那么就会有越来越多的大的 CELL 被当成整体被一次性访问, 而不是递归的一个一个访问每个孩子 cell。



图表 1: $\theta = 0.0$

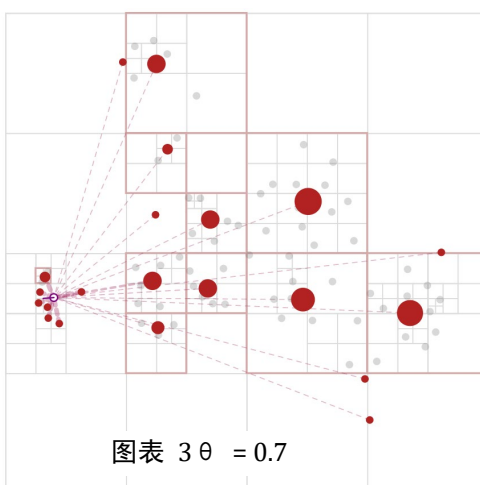
θ 等于 0 是一种比较极端的情况, 即每一个 cell 的边长和 sample point 到 cell 的距离的比值都会大于 0, 即都不是 long-range force, 所有要递归的访问每一个 cell 的每一个子 cell。现在我们改变 θ 的值, 观察, 有哪些 cell 被判定成了 long-range force, 这些 cell 就被当成整体, 用其质点计算和 sample point 之间的距离。



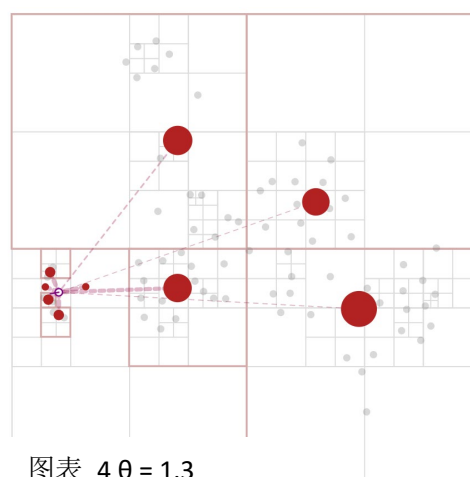
图表 2: $\theta = 0.50$

当我们将 θ 的值设置的大一点的时候, 存在一些 cell, 这些 cell 的边长与其到 sample point 的距离的比值小于该 θ (阈值), 这些 cell 将会被当做整体, 其质点将被访问。而哪些大于阈值的 cell 任然递归的访问所有的子 cell。

接下来持续的调大 θ 的值



图表 3 $\theta = 0.7$



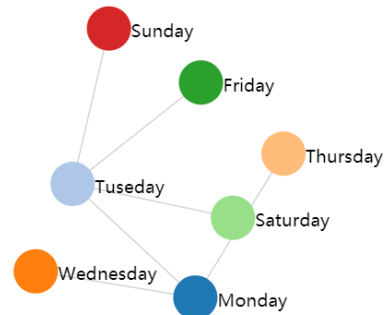
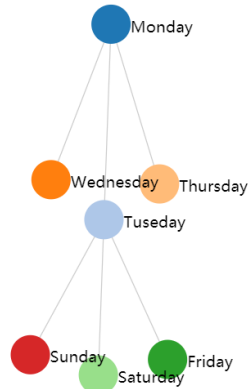
图表 4 $\theta = 1.3$

通过将 θ 的值逐渐的调大，我们发现，又越来越多的 cell 的边长和其到 sample points 的比值小于该阈值，那么这些 cell 对 sample point 的作用力都将被认为是 long-range force，所以这些 cell 将被当做整体，用其质心只计算一次其对 sample point 的作用力。

现在我们可以使用 Barnes-Hut 估计去限制总的比较的次数，并且该方法能够非常有效的加速。并且，当我们将 θ 的值设置的越大，那么速度就越快。并且一个较好的 θ 的取值应该是 1。

用 d3 构建 directed-force tree:

用 7 个节点来表示没一周的七个不同的天，然后拖动节点



可以发现力导向图的布局很好

结论分析与体会：

附录：程序源代码