PKU Visualization Blog 北京大学可视化与可视分析博客

Nanocubes: 对时空数据的事实探索 (Nanocubes for Real-Time Exploration of Spatiotemporal Datasets)

作者: Zhenhuang Wang 日期: 2013年11月11日

随着信息爆炸时代的到来，数据量越来越大，人们对时空数据的实时处理和探索显得越加困难。想象一下，假如你有一个微博数据集，它记录每条微博发布的时间、地点和发布设备。那么，你如何可以快速地知道到微博的地理分布呢，是上海还是北京的用户发的微博更多？人们是工作日里发的微博多还是周末发的多？每天微博发布的高峰时间是什么时候？人们用什么手机系统发的微博多呢，是iPhone还是Android？在2009年时候是什么情况呢？那么在2012年这种情况发生了变化吗？这些问题涉及到了各个维度上的聚合统计，并且在时间和空间维度还涉及到了不同的粒度。要回答这些问题，最简单的方法或许是扫描一遍数据集，然后获得统计值。但这在日益增长的数据量和实时性的要求下，这种方法显然不适用。

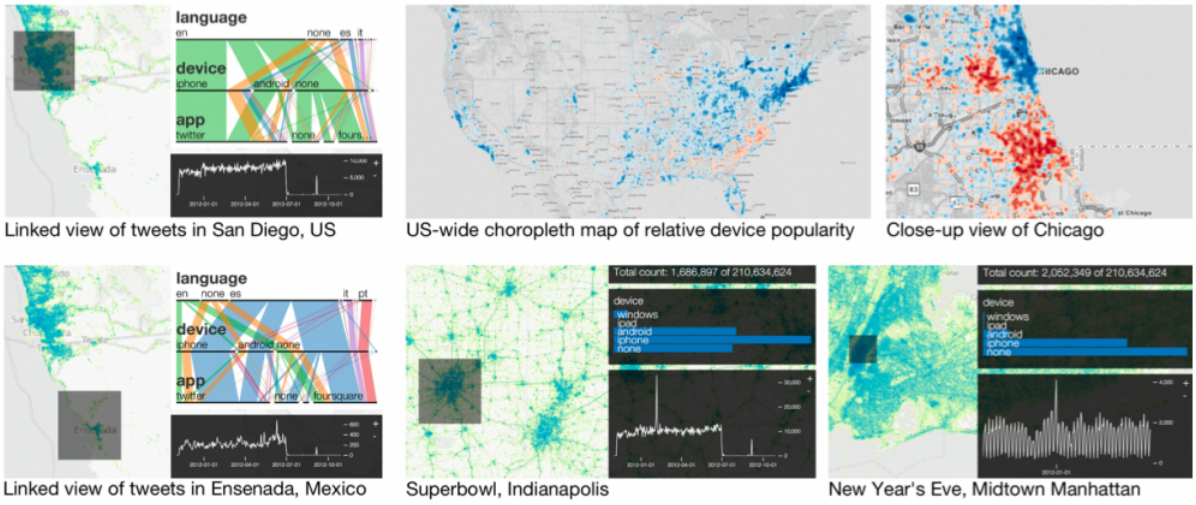

图1 使用nanocube对时空数据进行分析

为了支持多维度、多粒度时空数据的实时聚合分析，Lauro Lins等人发表的这篇文章中，在数据立方体的基础上提出了nanocube。简单地说，nanocube是一种数据结构，它可以对高维多粒度的时空数据进行高效的存储和检索。nanocube实际上是一种树结构，下面用一个例子简要说明nanocube的构造过程。

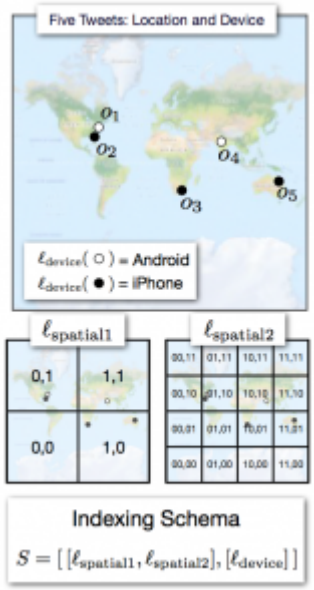

图2 示例数据

上图展示了一个有5个数据点的Twitter数据集，数据集包含了空间维度和一个属性维度：发布twitter的设备。其中，空间维度上分成两种不同的粒度。将上图展示的数据集整理成表格如下：

Object	Spatial1	Spatial2	Device
O1	0,1	01,10	Android
O2	0,1	01,10	iPhone
O3	1,1	10,01	iPhone
O4	1,0	10,10	Android
O5	1,0	11,01	iPhone

在构建nanocube时，数据点是一个一个被插入到树形结构中的。下图展示了每一个数据点插入到nanocube后的情况，其中插入新的数据点造成的变化用黄色背景标出。图中每一个维度之间用一条灰色横线隔开，可以看到最上面为空间维度，中间部分为设备维度，最底下则是各个数据点的索引。值得注意的是，在每个维度中，树的高度都等于该维度的层次数加1。例如，空间维度包含了2个层次的粒度，则在空间维度上，树的高度为3。树中的每一个节点都有一条边与下一维度的顶点相连。这样的目的是在检索的时候，可以不必到达最细的粒度，而可以在不同维度在不同层次上的聚合。例如如果只想搜索在[0,1]区域内使用Android设备的用户，那么只要在空间维度上进入[0,1]分支后即可直接进入设备维度，提高了检索效率。

这篇文章中，作者在六个测试数据集上使用了nanocube，并且通过密度图、时间线等可视化方法对数据集进行了可视分析。
下面这张图是Twitter平台上移动设备的分布情况，其中蓝色代表iPhone，红色代表Android。从时间线上不难

看出，Androdi呈上升趋势，并即将赶上iPhone。而地理空间上的分布来看，美国大部分区域依然是iPhone占据主流。不过如果我们细化到一些具体的城市，如芝加哥，就会发现在这些地方Android的覆盖率要高于iPhone。

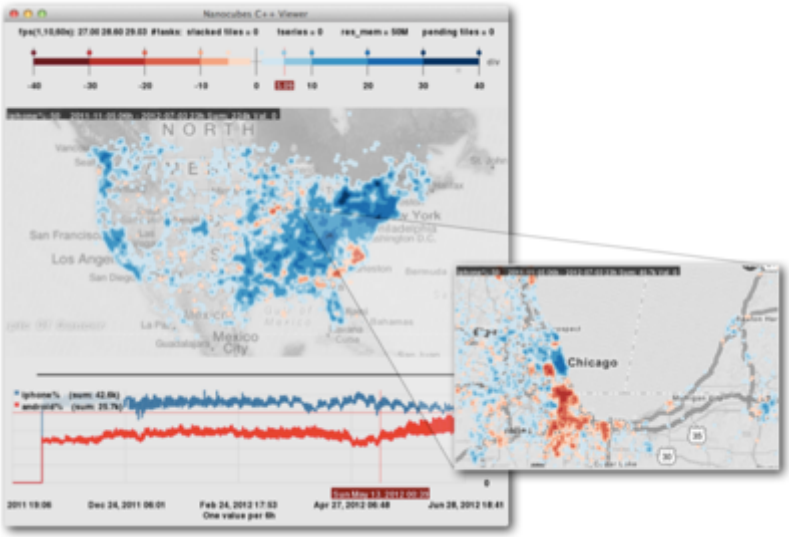

图3 Twitter数据分析

再例如，下图是电话记录。数据集中记录了每个电话的时间、位置和时长。从总体来看，人们在工作日会拨打更多地电话，而在周末电话数量会相对少些。但是在肯尼迪国际机场（JFK），工作日和周末的电话数量并没有明显的差别。然而，在曼哈顿这样的地方，周末电话数量减少的十分明显，大概工作在这儿的精英们周末都出去度假了吧！而在度假胜地拉斯维加斯，反而周末电话的数量要高于平日。另外，我们也可以观察电话时长的时间模式。通常来说，下午5点左右电话通信数量达到了一天中的高峰。但是如果我们只筛选出通话时间大于一小时的呼叫时，发现它们多是集中在午夜时分，这正是情人间、闺蜜间煲电话粥的黄金时段！

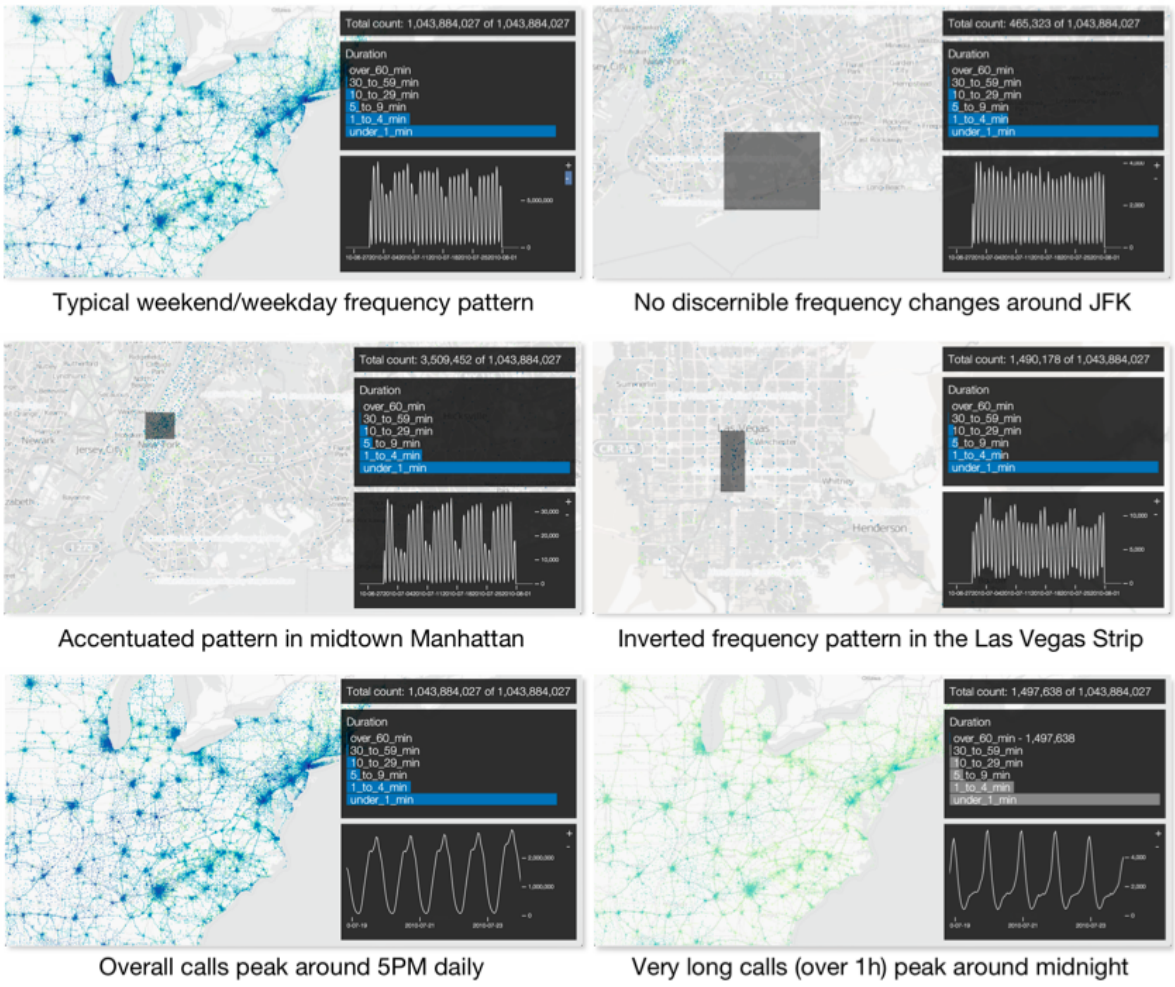

图4 通话数据分析

除了对数据集结果的分析，作者还对nanocube的时间、空间效率进行了分析。

dataset	objects (N)	memory	time	size	sharing	keys ($\|K\|$)	$\|K^\star\|$	schema
brightkite	4.5 M	1.6 GB	3.50 m	149.0 M	3.00x	3.5 M	2^{74}	lat(25), lon(25), time(16), weekday(3), hour(5)
customer tix	7.8 M	2.5 GB	8.47 m	213.0 M	2.93x	7.8 M	2^{69}	lat(25), lon(25), time(16), type(3)
flights	121.0 M	2.3 GB	31.13 m	274.0 M	16.50x	43.3 M	2^{75}	lat(25), lon(25), time(16), carrier(5), delay(4)
twitter-small	210.0 M	10.2 GB	1.23 h	1.2 B	3.72x	116.0 M	2^{53}	lat(17), lon(17), time(16), device(3)
twitter	210.0 M	46.4 GB	5.87 h	5.2 B	4.00x	136.0 M	2^{60}	lat(17), lon(17), time(16), lang(5), device(3), app(2)
splom-10	1.0 B	4.3 MB	4.13 h	51.2 K	5.67x	7.4 K	2^{20}	d1(4), d2(4), d3(4), d4(4), d5(4)
splom-50	1.0 B	166.0 MB	4.72 h	8.8 M	16.00x	1.9 M	2^{30}	d1(6), d2(6), d3(6), d4(6), d5(6)
cdrs	1.0 B	3.6 GB	3.08 h	271.0 M	18.60x	96.3 M	2^{69}	lat(25), lon(25), time(16), duration(3)

图5 Nanocube的时间、空间效率分析

从上面的表格中可以看到，对于上百万、甚至上亿的数据集，nanocube依然可以在普通笔记本电脑的内存中完成索引，空间利用率十分高。
如果你手头也有类似的聚合需求，不放尝试使用一下nanocube吧！

参考文献

[1] Lauro Lins, James T. Klosowski, Carlos Scheidegger. Nanocubes for Real-Time Exploration of Spatiotemporal Datasets. IEEE Transactions on Visualization and Computer Graphics (InfoVis '13), 19(12), pp. 2456-2465, 2013.

论文报告　　多层次, 多维度, 数据存储, 数据立方体, 时空数据