Report 9/23

林涛

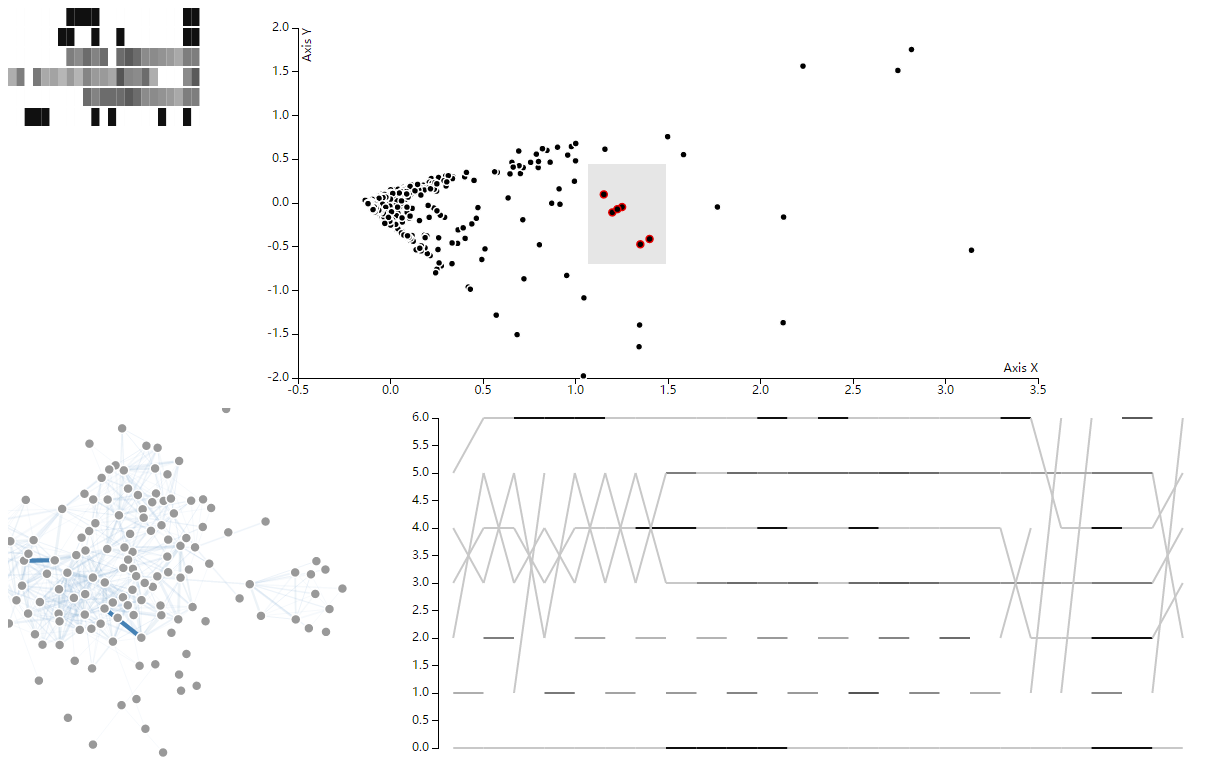
# 使用类似DAG中的方法去交叉

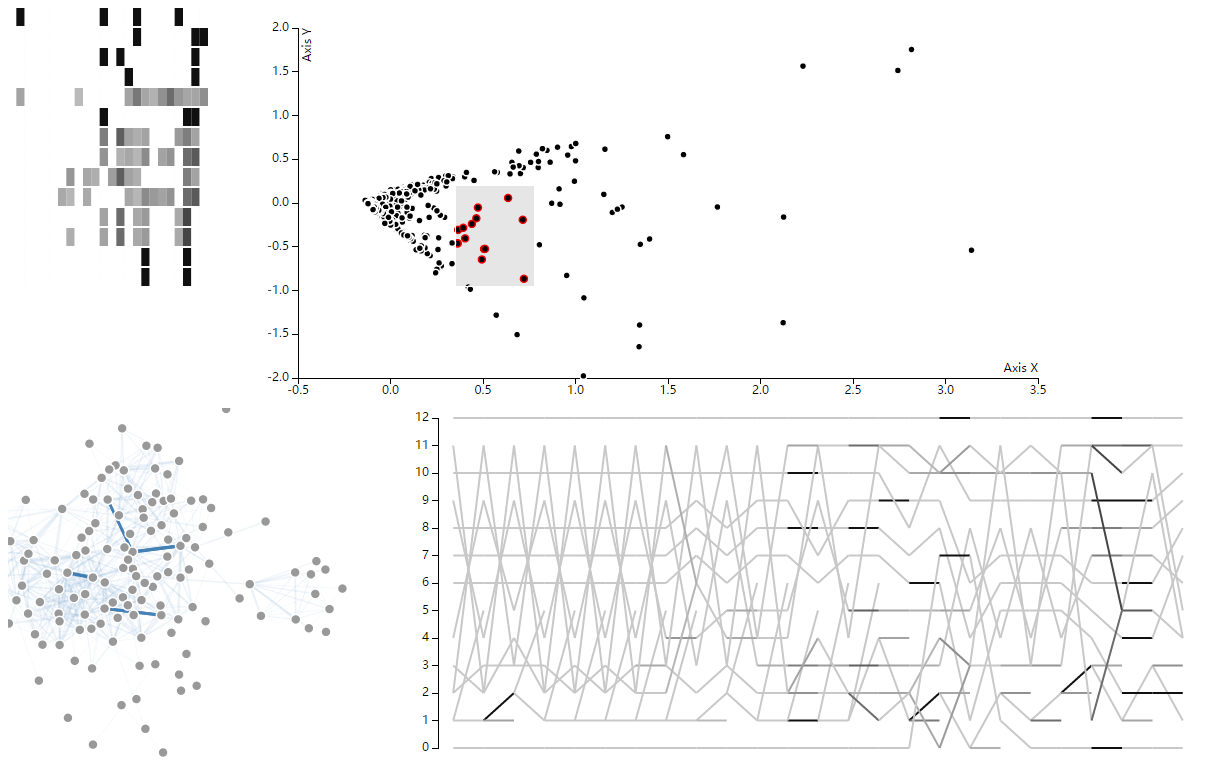
首先的一个与之前不同的地方是：之前是把所有的node拿出来排，现在是只将选中的edge相关的node参与后面的排列和显示。因此在空间的利用上会比之前充裕一些。

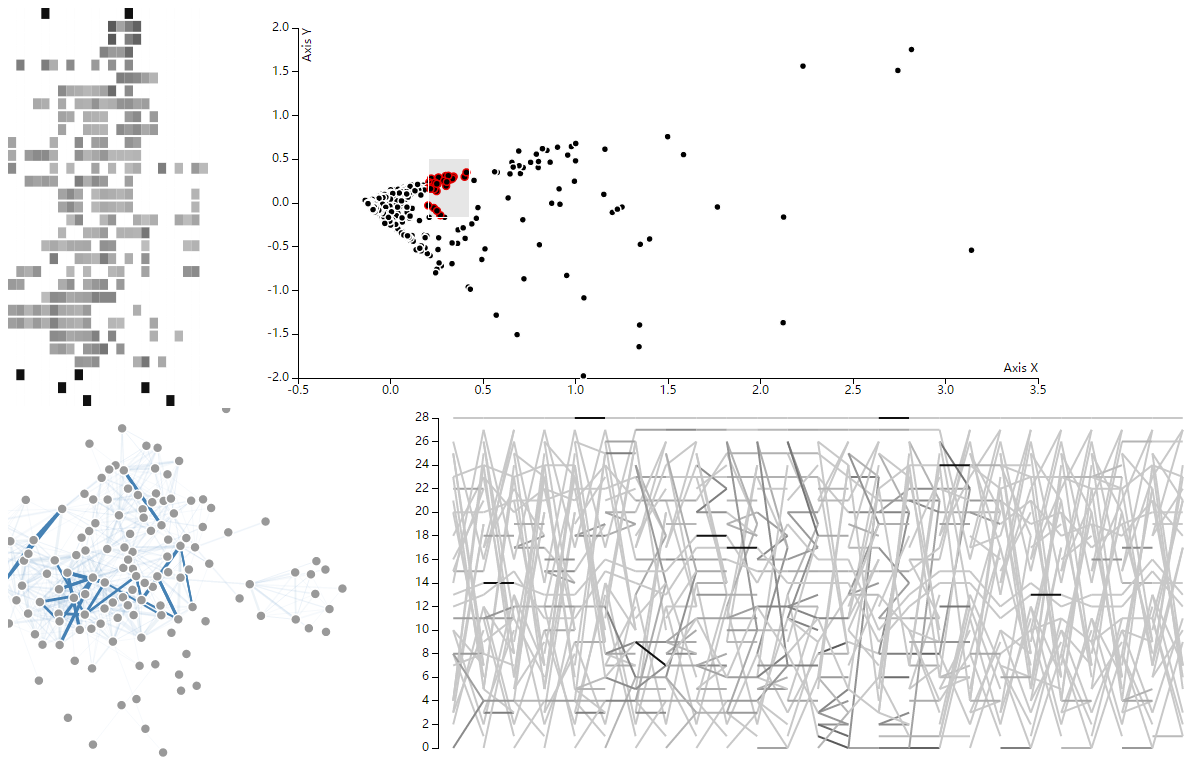
使用Sugiyama Method中的第三步:去交叉部分。去交叉的算法使用The Median Method。

第一排的node是随机suffle的。之后每排的顺序都是根据前一排计算出来。

计算的方法是：找要排的这个节点对应的上一排中连过来的点的位置的中位数。如果这个位置已经填了，就往附近放。放完这一排中与上一排有连接的点后，剩下的点随便填满。

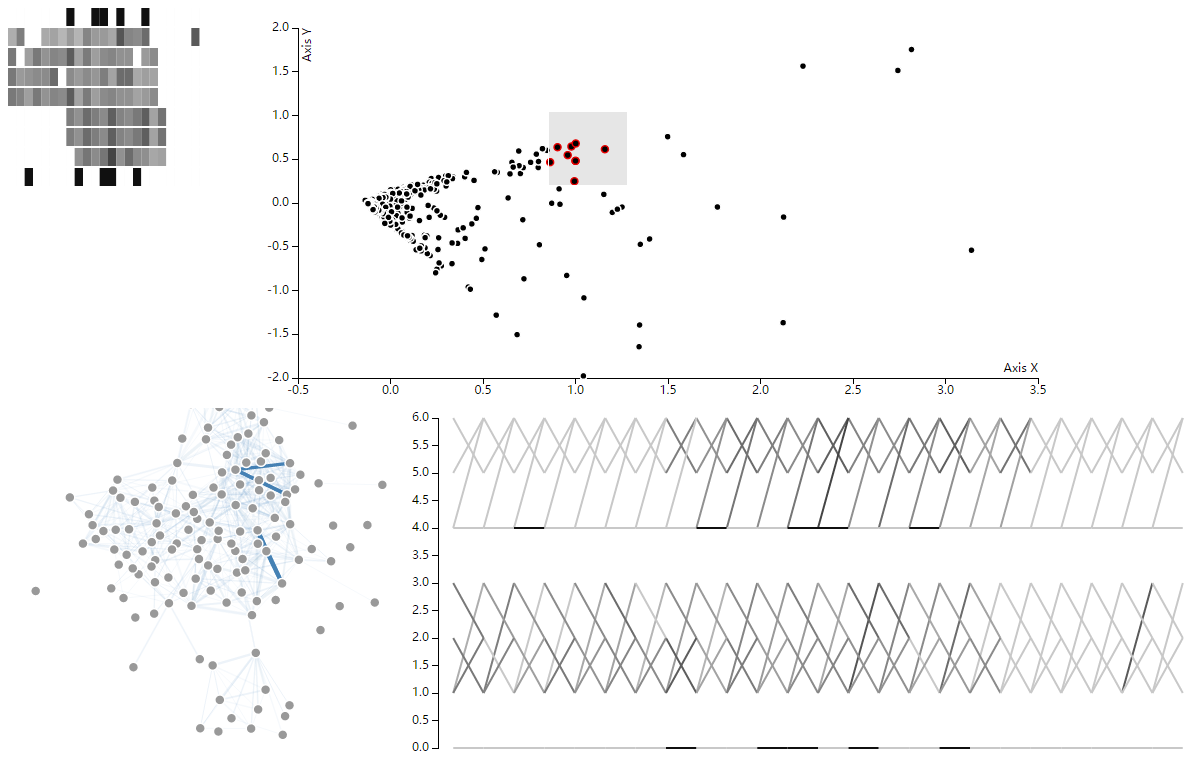






因为按照这个思路作，每一排node的顺序是不一样的，甚至很难在bipartite view上看清楚哪里是同一个edge（同一个edge的线段并不是连起来的，因为这个edge是有向的，对每一条线段来说左边的node是出发的点，右边的node是到达的点，不会首尾相连）。

我觉得如果要让这个视图真正起作用的话，每一栏的node顺序应该统一。这样分析者至少能通过线段的平行。这个图是统一node顺序的（但没有使用去交叉）：



如果要在统一node顺序的情况下去交叉，就不能使用前面提到过的算法了。我最近在思考有什么方法，但还没想到效率上说得过去的。我觉得应该有人做过这样的问题，但找了找文献没找到。

# 其他

* 更改了颜色映射
* 加粗了bipartite view中的线条

# 接下来的工作

* 使用erone的数据
* 用连接次数作为边的值
* 完善交互和界面