**发现团伙算法文档**

1. 权重计算：
2. 对每一属性，假设两个UID：A与B，关于该属性轻重得分分别为a1,a2与b1,b2，则a = a1+2\*a2,b=b1+2\*b2，该属性对A,B节点之间边的权重贡献值为：1/（1/a+1/b），进而得到关于每个属性的权重矩阵W0。
3. 用W0表示关于某种属性的权重矩阵,A,B分别为第i个与第j个uid，则w\_ij = w\_ji，表示第i个节点与第j个节点之间的边的总的累计权重。并通过函数：f(x) =(1-exp(-s\*x))/(1+exp(-s\*x))，将w\_ij映射至区间[0,1），得到归一化后的权重矩阵W0。其中s = log(2)/median(w\_median)，这里w\_median为图G中所有边的权重值（不为0）的中位数。最终再将5个归一化后的权重矩阵乘以每种属性对应的权重值相加，得到最终的权重矩阵W（身份证属性对应权重为0.4，车牌号与银行卡对应权重为0.2，手机号与设备对应权重为0.1）。
4. 将W权重矩阵的第i行所有元素相加，得到第i个节点的度数D\_i（第i个节点所有相邻的边的权重之和）。
5. 模块度计算：
6. 计算整个图G所有边的权重之和，记为m。
7. 设图G此时被分为了N个团体，第i个团体内部连接的边的权重和为w\_i，第i个团体所有节点在图G中的度数之和为d\_i，则第i个团体对图G模块度的贡献值q\_i = w\_i/m-(d\_i/m)^2。
8. 图G模块度Q则为所有q\_i的和。
9. 算法迭代：
10. 首先把每个独立的节点当作一个团体，计算得到最初始图G的模块度Q。
11. 按编号顺序，从第一个点开始遍历，对每一个点，首先计算该点从目前的团体X中出来后带来的模块度变化量delta\_q（该点出来后，团体X的某块度减去该点在团体X中时，团体X的模块度的差值）。用X1,X2,…,Xn表示其余所有与该点相邻的团体。依此计算当该点加入团体Xi后带来的团体Xi的模块度变化量delta\_qi（该点加入团体X\_i后，团体X\_i的模块度减去该点加入之前，团体X\_i的模块度）。取delta\_q+delta\_qi(i=1,2,…,n)的最大值，记作delta\_Q。设delta\_Q = delta\_q+delta\_qj。若delta\_Q>0，则讲该点归属于团体Xj，若delta\_Q<=0，则该点依然在之前所在团体不变。
12. 遍历后计算每一次遍历所属团体发生变化的点的个数M,如果M>0，则回到上述第2步，直到M=0为止，再转入第4步。
13. 将同一个团体的所有点合并为同一个点，构建新的图G，新的点所自带的边（简称：环）的权重为合并前该团体内部所有边的权重之和，新的图G中任意2点之间边的权重等于合并前2点对应的2个团体之间所有边的权重之和。新的图G中每个点的度数，等于该店所有相邻边的权重之和（如果是自带的边的权重，计算度数时要算2次）。
14. 对新的图G通过第2步的方法进行迭代，如果第1次迭代中发生变化的点个数M=0，则整个算法停止，当前分团即为最终结果。否则重复2，3，4的步骤，直到迭代结束为止。