<http://blog.csdn.net/zouxy09/article/details/49105265>

<http://scikit-learn.org/stable/modules/label_propagation.html>

这里两篇文档都是在说LPA算法用于半监督学习，所以需要构建基于两两相似度的图。这个构建本身就是需要nn的复杂度

2、基本思想   
标签传播[**算法**](http://lib.csdn.net/base/datastructure)是不重叠社区发现的经典算法，其基本思想是：将一个节点的邻居节点的标签中数量最多的标签作为该节点自身的标签。给每个节点添加标签（label）以代表它所属的社区，并通过标签的“传播”形成同一标签的“社区”结构。

给每个节点添加标签（label）以代表它所属的社区，并通过标签的“传播”形成同一标签的“社区”结构。一个节点的标签取决于它邻居节点的标签：假设节点z的邻居节点有z1至zk，那么哪个社区包含z的邻居节点最多z就属于那个社区（或者说z的邻居中包含哪个社区的标签最多，z就属于哪个社区）。优点是收敛周期短，无需任何先验参数(不需事先指定社区个数和大小)，算法执行过程中不需要计算任何社区指标。

时间复杂度接近线性：对顶点分配标签的复杂度为O(n)，每次迭代时间为O( m)，找出所有社区的复杂度为O (n +m)，但迭代次数难以估计

3、传播过程：   
1)初始时，给每个节点一个唯一的标签；   
2)每个节点使用其邻居节点的标签中最多的标签来更新自身的标签。   
3)反复执行步骤2)，直到每个节点的标签都不再发生变化为止。   
一次迭代过程中一个节点标签的更新可以分为同步和异步两种。所谓同步更新，即节点z在第t次迭代的label依据于它的邻居节点在第t-1次迭代时所得的label；异步更新，即节点z在第t次迭代的label依据于第t次迭代已经更新过label的节点和第t次迭代未更新过label的节点在第t-1次迭代时的label。

注：   
1、迭代次数设定一个阈值，可以防止过度运算；   
2、对于二分图等网络结构，同步更新会引起震荡；   
//3、类似（“强”社区>）定义的结构（该社区>=）；   
4、每个顶点在初始的时候赋予唯一的标签，即“重要性”相同，而迭代过程又采用随机序列，会导致同一初始状态不同结果甚至巨型社区的出现；   
5、如果能预测“社区中心”点，能有效提高社区发现的准确度，大幅提高效率；   
6、同一节点的邻居节点的标签可能存在多种社区最大数目相同的情况，取“随机”一个作为其标签

4、算法改进思路：初始化或传播改进   
1)给节点或边添加权重（势函数、模块密度优化、LeaderRank值、局部拓扑信息的相似度、标签从属系数等），信息熵等描述节点的传播优先度，进而初步确定社区中心点以提高社区划分的精度；   
2）标签初始化改进，如提取一些较为紧密的子结构来作为标签传播的初始标签（非重叠最小极大团提取算法 orz。。。）或通过初始社区划分算法先确定社区的雏形再进行传播。   
3）标签随机选择改进，将1）中的权值和节点邻接点的度数等作为参考因素，对标签更新过程进行修正。

1)在社区中寻找不重叠三角形作为起始簇的雏形，以提高算法结果的稳定性和运行效率；   
2)添加标签熵属性，在迭代过程中不采用随机序列，而是根据每个节点的标签熵来排序序列；   
3)在2）的基础上，为了不完全消除标签传播算法的随机性，将排序好的队列平均分成三个部分，在每个部分内，节点进行随机排列。   
4)对于同一节点的邻居节点的标签可能存在多种社区最大数目相同的情况，不使用随机方法，而是分析该节点的邻节点的邻节点集标签分布情况来决定该节点的标签   
5)在社区中寻找以度最大的若干节点为中心的“雪花型”结构作为起始簇的雏形   
在实现的过程中，将上述方案进行组合衍生出更多的可行方案，初步试验结果表明算法的随机性与稳定性很难同时保证，设定起始簇的结构收敛速度快但有可能生成巨型社区；在节点较少的情况下，标签熵的方法准确率和稳定性最好；至于组合方案初步的试验验证发现效果反而下降了。