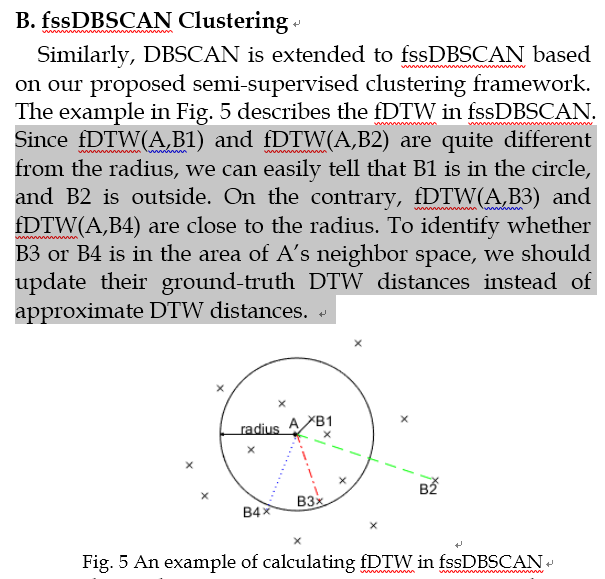
**蒋文君周总结(20190728)**

1. 协助修改潘彦州学长的论文；
2. 总结理解学长的论文，用python重写了学长论文中的部分代码，加深理解。学长的论文提出了STSC(半监督时间序列框架)：主要包括一种快速的相似度衡量和成对约束的传递策略。基于此设计了fss-kmeans和fss-DBSCAN两种算法，对11种数据进行了实验测试。下面是对该论文的一点总结和不解：

（1）快速的相似度衡量的核心思想是：通过计算两个样本的fDTW值(时间复杂度为O(n)),若两个样本的fDTW值相差值大于参数δ，则进一步计算两者的DTW值（时间复杂度为O(n2)），比纯粹计算比较DTW值更节省时间。由于代码文件里只有对kmeans算法的改进，并没有看到fDTW是怎么拓展到DBSCAN里的具体方法。论文中的说法是这样：



对于这个做法我不是很理解，我以为只要比较其他样本点（B1,B2,B3,B4）距中心点（A）的距离是否小于半径即可，小于半径即在圈内。没有必要对比B1,B2,B3,B4之间的距离，那么fDTW也就不需要使用。

（2）成对约束：就是有很多类似[p1,p2]这样的点对，若在mustlink里，该点对则表示p1和2必属于同一类；反之，在cannotlink里该点对表示p1和p2必不属于同一类。像这样的约束条件一般来自少量已有标签的数据，用于半监督分类中以提高分类效果。

（3）参考学长代码，最终得到fss-kmeans的python代码，因为memory error的问题，还没有调通。另外，论文在描述fss-kmeans的算法中提到下面两步，但是代码中并没有体现：

S ← Build a few core groups with pairwise constraints P

← Construct an updated dataset based on solution 1 and 2

1. 结合之前看过的论文，总结了：

（1）聚类评价方法：Purity、RandIndex和NMI，并用python实现其代码。

（2）相似距离计算方法：欧式距离(ED)、DTW、LB\_Keogh、LB\_enhanced及fDTW。