**t-SNE降维算法学习总结**

t-SNE(t-distributed Stochastic Neighbor Embedding)是一种将高维数据降维到二维或三维的非线性降维方法，它由Maaten与Hinton于2008年提出\*，是基于Hinton在2002年所提出的随机近邻嵌入(SNE)方法改进而来，是一种适用于数据可视化的降维算法。

1. **预备知识**

KL距离：即Kullback-Leibler差异(Kullback-Leibler Divergence)的简称，也叫做相对熵(Relative Entropy)，是衡量相同事件空间中两个概率分布的差异情况，表述如下：

当两个概率分布相同时，其KL距离为0。

概率分布的信息熵为，即编码时，平均每个事件至少需要的比特数，由此上式可解释为在概率的空间，以表示平均每个事件编码增加的比特数。需要注意的是，KL距离不同时满足距离的三个条件：非负性、对称性、三角不等式，并非典型的距离定义。

***t*分布：**又称学生*t*分布，即某正态总体的均值为，方差为，在该总体中随机抽取容量*N*的随机样本，它们的均值为，方差为，则随机变量*t*为：

此时*t*服从自由度为的*t*分布，即，需要说明的是，自由度越大，*t*分布越趋向于正态分布，其分布图线的尾部两端较为平缓，是一种长尾分布，该分布在处理小样本和异常点时具有一定优势。

1. **SNE算法**

降维是将数据点在高维空间的相似性特征保持到低维空间中，一般是以欧式距离作为相似性度量。而SNE算法是将距离关系转化为概率表示，以概率表征降维前后的数据特征。

假定高维空间中的两点与，以条件概率选择作为其邻近点，以正态分布定义：

其中为以为中心点的正态分布方差，且定义。

与在降维对应的低维空间中映射为与。同样地，定义低维空间中的条件概率，为简便公式，定义低维空间中正态分布方差为：

同样地，。

考虑到其它所有点与的概率关系，则以表示，同理以表示低维空间中其它所有点与的概率关系，如果能够较好地保持两个空间的点关系，则与近似。我们采用KL距离表示这种相似性，SNE的优化目标就是最小化所有数据点的KL距离，其代价函数如下：

使用梯度下降法对上式求梯度，有：

变换上式，可得最终迭代公式如下：

其中为迭代步长，为第t次迭代冲量，并假设初始随机化的低维点集为。

但是由于KL距离不具有对称性，由代价函数可知，当较大时，较小，C较高，反之C较小。这一现象在空间中可以描述为若高维空间中两点距离较近，如果映射到低维空间中距离较远则应加以一个较高的惩罚，符合情理；但是若映射到低维空间中距离较近则要加以一个较低的惩罚，这就有违常理，理应加以一个较高的惩罚。因此SNE的代价函数仅关注局部结构，而忽略了全局。

1. **t-SNE算法**

SNE的主要问题是它的不对称行度量。在t-SNE算法中，第一步是修正SNE为对称的。即分别在高维和低维空间定义和：

使得，，以此消除不对称性，同时也消除了离群点惩罚值不高的情形。其中n为数据点数目。以P与Q分别表示在高维和低维空间构造的联合概率分布，代价函数更新如下：

梯度更新如下：

由此解决了SNE中的不对称问题。但是SNE还存在拥挤问题(Crowding Problem)，也就是在SNE降维后不同类簇叠加无法区分，这是高维空间分布和低维空间分布的差异造成的。通俗地说，十维空间存在11个两两间距相等的点，二维空间仅有3个点满足条件，因此高维空间点的关系无法完全保留到低维空间。

为了解决拥挤问题，在高维空间使用正态分布将距离转化为概率，在低维空间使用*t*分布将距离转化为概率，使高维度中小距离点在映射后能够拉开间距，也就是在低维空间中同类簇点聚合的更紧密，不同簇间点更加疏远。同时t分布的长尾较高，对异常点不敏感，具有较高鲁棒性，拟合更为合理。

以自由度为1的*t*分布改写：

梯度改写为：

以上即为t-SNE算法，

其优点：

• 规避了SNE算法中的不对称和拥挤问题，使不同簇间的点在低维空间中分离开；

• 由于*t*分布的长尾特性，减小了离群点对于概率分布模型的影响。

缺点：

• 时间复杂度较高；

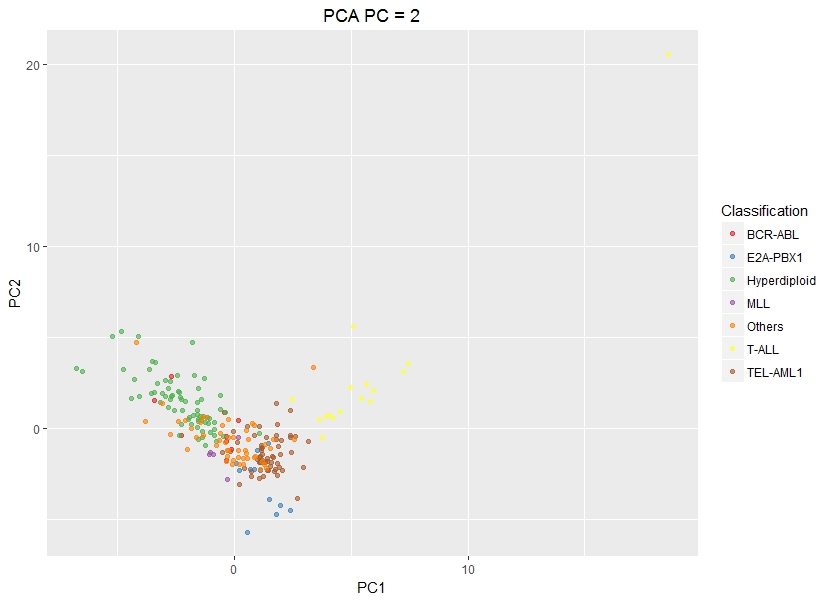
• 过程中所应用的概率模型以当前数据集得出，无法适应到其他数据集中测试，仅能应用到特定数据的二维或三维的可视化；

• 算法中的距离是没有意义的，若匹配基于距离的机器学习算法中会存在较大偏差。

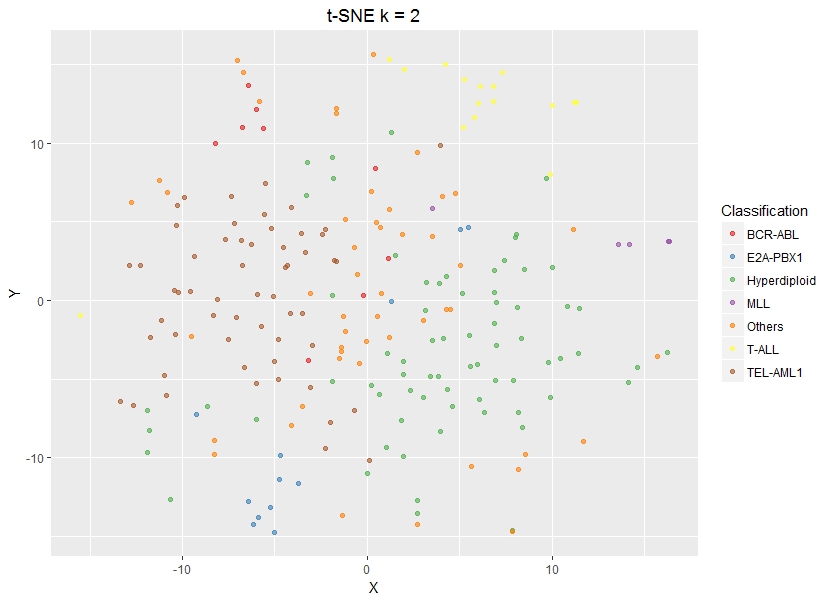
1. **实验探究**

以规范化后的Samples219数据集(样本数219，特征维度60，类别数7)试验，将使用t-SNE方法与PCA在二维空间的降维效果进行对比。

PCA图示：

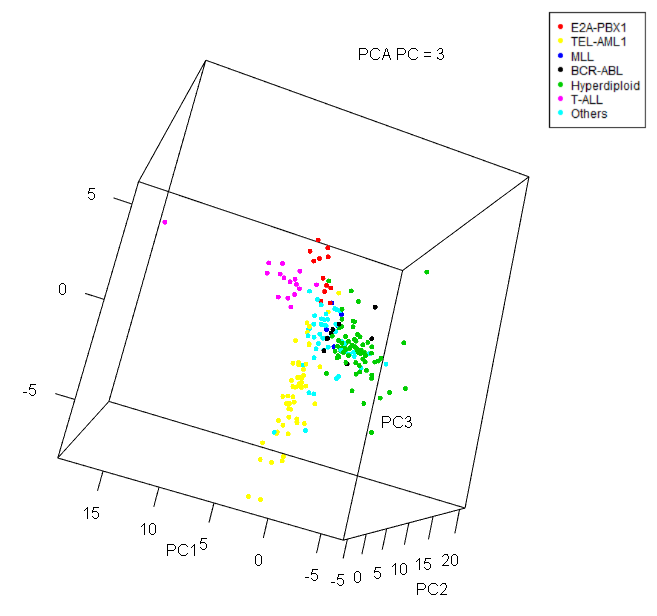


t-SNE图示：

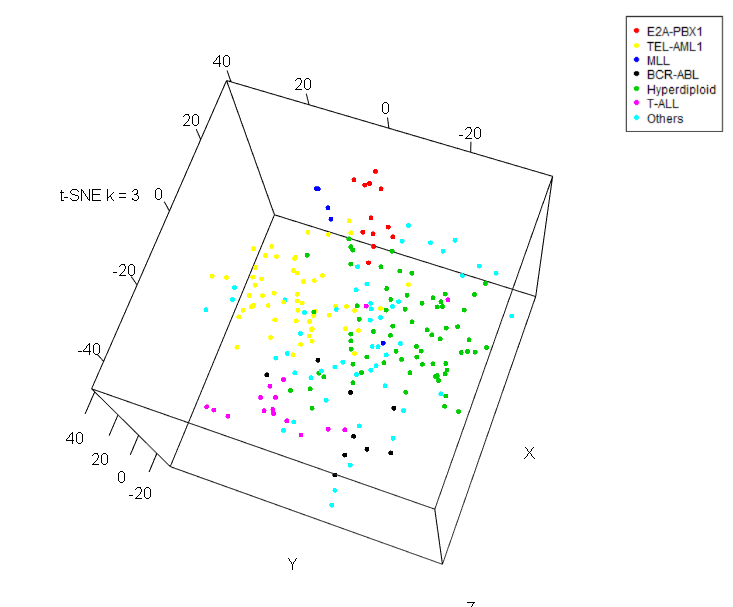


三维空间降维效果如下：

PCA图示：



t-SNE图示：



经实验，无论是在二维还是在三维，t-SNE算法的效果均优于PCA，不同类簇的间距得以拉开，而t-SNE算法的三维图示的效果要优于二维图示，所保留的原始信息更多。