大作业：单细胞测序

钟清扬　自65　2016011481

1. 细胞原始数据分类

数据集中共有1000个细胞数据，其中有"H1"类58个，"GM"类211个，"K562"类380个，"TF1"类43个，"HL60"类63个，"BJ"类61个，"Leuk"类23个，“LSC1”类27个，“Blast”类25个，”LSC2“类21个,“LMPP”类39个，”mono“类49个. 每个组织具有239255个特征,属于超高维数据，因此首先需要进行特征筛选和降维。

实验中尝试了3种特征选择或降维方法与4种分类方法，通过5折交叉验证对各模型的参数进行选择，并最终选择合适的特征筛选或降维方式与合适的分类器对测试集标签进行预测，此部分代码详见`HMW1.py`

实验中首先对数据进行预处理，将细胞名称处理为0~11的数字形式以便进一步分类。

(1)特征筛选与降维

过滤法先选特征再学习，通过设计某种统计量来度量特征x和响应值y之间的关联程度，选出其中“关联程度”最高的k个特征或者高于阈值t的所有的特征。常见的过滤法有基于类内类间距离的判据、基于随机变量关联性的判据，基于熵的可分性判据，基于概率分布的特征筛选，用统计验证作为可分性判据等等。

a.相关系数法

相关系数法将特征取值X和响应值Y（类别标签）都视作随机变量以衡量二者之间的关联程度

$$corr(X,Y)=\frac{\sum_{i=1}^{n}(x_i^{(k)}-\overline{x_i})(y_i^{(k)}-\overline{y_i})}{\sqrt{\sum_{i=1}^{n}(x_i^{(k)}-\overline{x_i})^2\sum_{i=1}^{n}(y_i^{(k)}-\overline{y_i})^2}}$$

代码具体实现如下：

```
Corr=[]
length=x_data.shape[1]
for i in range(length):
    cor=np.corrcoef(x_data[:,i].T,y_data)
    Corr.append([i,abs(cor[0][1])])
Corr.sort(key=takeOrder,reverse = True)
Corr=np.array(Corr)
featureList=Corr[0:100,0].astype(int)
print("correlation coefficient feature:",featureList)
```

通过对相关系数绝对值进行排序，得到前10维最显著的特征为12733，58230，118483，118882，14595，315，18524，333，118812，319。将所选特征的列进行拼接，即可得到特征选择后的新训练集数据。实验发现选取前1000维特征时可以取得较好的分类效果。

b.方差阈值法

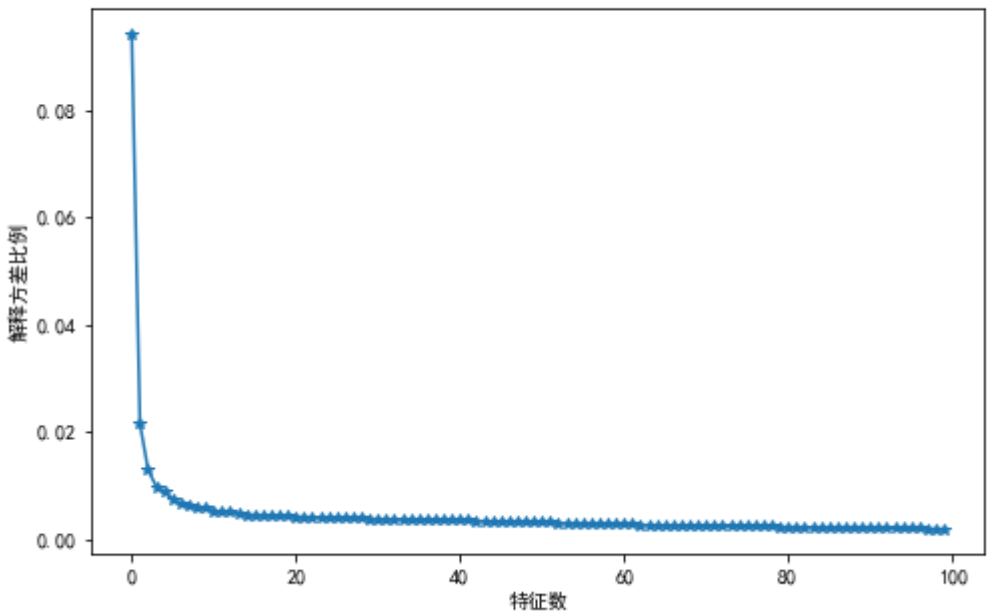
方差阈值是特征选择的一个简单方法，通过去掉所有方差没有达到阈值的特征获得降维后的训练集。代码具体实现如下：

```
def VarianceThre(x_train,x_test):
    sel = VarianceThreshold(threshold=0.8)
    x_train_new=sel.fit_transform(x_train)
    x_test_new=sel.transform(x_test)
    print(x_test_new.shape[1])
    return x_train_new,x_test_new
```

设置训练集方差阈值为0.98时，所利用特征数在750维左右，可以取得较好的分类效果。

c.PCA特征变换

设定降维至100维，前100维中各维特征所解释的百分比如下：




```
from sklearn.decomposition import PCA
def PCA_Reduction(x_train,x_test):
    #pca1 = PCA(n_components=0.98)
    pca1 = PCA(n_components=100)
    pca1.fit(x_train)
    x_val_new = pca1.transform(x_val)
    x_train_new = pca1.transform(x_train)
    return x_train_new,x_val_new
```

PCA属于特征变换而非特征选择，将原数据变换为100维正交向量进行表示时可以取得较好的分类效果

(2)分类器设计与选择

a.支持向量机法

使用支持向量机时，可以设置参数decision_function_shape='ovr'，每次选取一个类别与其他类别进行划分以实现多分类任务；也可将参数设置为'ovo'，对类别两两进行划分，用二分类的方法模拟多分类结果。

根据核的不同，支持向量机是通用的线性、非线性分类器，具有良好的理论支撑，但训练数据过多时会导致计算开销过大。由于支持向量机的拟合时间复杂度大于样本数的二次方，因此支持向量机更适用于数据规模较小时的情景。支持向量机所选用的核函数越复杂消耗的时间越长。由于支持向量机的分类效果与核函数的选取密切相关，实际使用时还要选择合适的核函数以取得较好的分类效果。

采用方差阈值法与支持向量机结合时，5折交叉验证的平均正确率为75.3%；采用相关系数法选取特征时，支持向量机的平均正确率为83.7%；采用PCA降维与支持向量机结合时，交叉验证的平均正确率为94.7%。

```
from sklearn.svm import SVC
from sklearn.linear_model.logistic import LogisticRegression
def SVM_Classify(x_train,y_train,x_test,y_test):
    clf = SVC(kernel='linear',verbose=1)
    clf.fit(x_train, y_train)
    y_predition_test=clf.predict(x_test)
    total=0
    right=0
    for i in range(len(y_predition_test)):
        if y_predition_test [i]==y_test[i]:
            right+=1
        total+=1
    acc=float(right/total)
    print('SVM val accuarcy: ' + str(acc))
    return acc
```

b.Logistic Regression

多元逻辑回归常见方法有OvR和MvM两种，一般而言MvM分类相对精确。Logistic Regression是线性分类器，具有简单、可解释性强等优点，但适用范围较有限。

采用方差阈值法与Logistic Regression结合时，平均正确率为78.9%；采用相关系数法与Logistic Regression结合时，平均正确率为86.7%；采用PCA降维与Logistic Regression结合时平均正确率为94.4%.

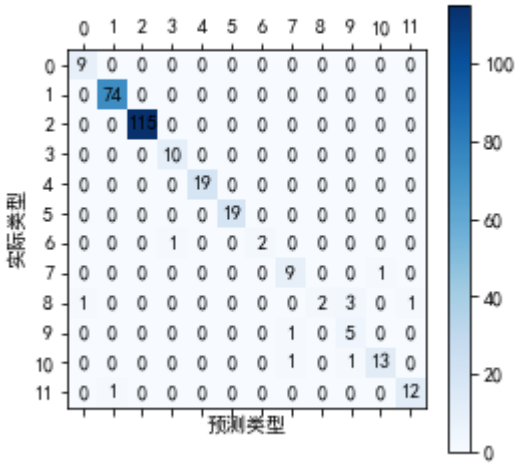
```
def Logistic_Regression(x_train,y_train,x_test,y_test):
    classifier=LogisticRegression()
    classifier.fit(x_train,y_train)
    y_predict=classifier.predict(x_test)
    total=0
    right=0
    for i in range(len(y_predict)):
        if y_predict[i]==y_test[i]:
            right+=1
        total+=1
    acc=float(right/total)
    print('Logistic Regression val accuarcy: ' + str(acc))
    return acc
```

c.神经网络

神经网络是通用的非线性分类器，功能强大，能处理多类问题；但是计算开销大，易出现过拟合或陷入局部最优解，存在模型可解释性差等问题，模型效果不稳定。

采用方差阈值法与隐层节点数为100的单隐层神经网络结合时，平均正确率为57.7%；采用相关系数法进行特征选择时，神经网络的平均正确率为89.1%；采用PCA降维时平均正确率为90%.

采用PCA降维的方法与隐层节点数为100的单隐层神经网络结合时，个别情况下分类效果会得到很大提升，验证集正确率约为96.33%，混淆矩阵如下：



d.随机森林方法

随机森林算法属于集成学习算法，包含多个决策树分类器。通过对各个树的预测结果进行汇总投票以作为最终预测，可以有效提升模型准确性；此外随机森林训练过程并行，因而训练速度可以很快。在解决大规模数据与高维度数据时，随机森林可以取得更好的效果。

首先可先利用GridSearch选取最优的不纯度准则、最大树深和最小不纯度。实验发现选取gini不纯度、最大树深为120，最小不纯度为0时可以取得最好的分类效果。

采用方差阈值法与随机森林结合时，5折交叉验证的平均正确率为53%；采用相关系数法选取特征时，随机森林的平均正确率为64%；采用PCA降维与随机森林结合时，交叉验证的平均正确率为93%。

```
def RandomForest(x_train_new,y_train,x_val_new):
    dtree=RandomForestClassifier(criterion='gini',max_depth=120,
                                 min_impurity_decrease=0)
    dtree.fit(x_train_new,y_train)
    pred=dtree.predict(x_val_new)
    print(classification_report(y_val,pred))
```

(3)测试集预测

通过对各种特征筛选或降维以及分类的方法进行尝试，发现采用PCA降维后模型分类效果更好且表现较为稳定。为进一步提高预测效果，实验中采取类似于Adaboost算法的做法，每次随机选取80%的数据对采用PCA方法降维的模型进行训练与预测，每个模型均预测5次。将所有模型的预测结果进行投票，将出现次数最多的标签作为测试集最终的标签

具体代码实现如下：

```
prediction_list=[]
for i in range(5):
   x_train, x_val, y_train, y_val = train_test_split(data[:,:-1], data[:,-1],
test_size=0.2)
   x_train_new,x_test_new =PCA_Reduction(x_train,x_test)
   print(x_test_new.shape)
   #支持向量机
   clf = SVC(kernel='linear',verbose=1)
   clf.fit(x_train_new, y_train)
   y_predition_test=clf.predict(x_test_new)
   prediction_list.append(y_predition_test)
   #LogisticRegression
   classifier=LogisticRegression()
   classifier.fit(x_train_new,y_train)
   y_predict=classifier.predict(x_test_new)
   prediction_list.append(y_predict)
   #随机森林
   dtree=RandomForestClassifier(criterion='gini',max_depth=120,min_impurity_decrease=0)
   dtree.fit(x_train_new,y_train)
   pred=dtree.predict(x_test_new)
   prediction_list.append(pred)
prediction_result=np.array(prediction_list).T
test_result=[]
for x in prediction_result:
    test_result.append(np.argmax(np.bincount(x)))
```

预测最终结果在`problem1.txt`中.

2.依据调控因子分类

(1)特征筛选与降维

依据调控因子进行分类时数据集中仍有1000个细胞数据，但细胞数据已依据调控因子进行了预处理。每个组织受870个调控因子的调控，数据维数明显降低。除尝试问题1中的特征筛选与降维方法外，此处还尝试了利用sklearn自带的Univariate feature selection过滤法与包裹法对特征进行选择。

a.过滤法选取特征

使用sklearn中集成的模块SelectKBest以实现过滤法特征选择，具体代码实现如下

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
def filtering(x_train,y_train,x_val):
    model=SelectKBest(f_classif,k=200)
    label=model.fit(x_train,y_train).get_support(indices=True)
    x_train_new=model.transform(x_train)
    x_val_new=x_val[:,np.transpose(label)]
    return x_train_new,x_val_new
```

选取200个特征时可以取得较好的分类效果，所选取的特征维数为：2, 20, 37, 41, 47, 55, 66, 68, 70, 74 .....

b.包裹法选取特征

除过滤法外，sklearn中有两种递归式特征消除的方法以实现包裹式特征选取。

RFE 通过学习器返回的 coef_ 或者 feature_importances_ 属性获得每个特征的重要程度后，将每次从当前特征集合中移除最不重要的特征，不断重复递归步骤直到最终达到所需要的特征数量为止。RFECV则通过交叉验证来找到最优的特征数量。若减少特征会造成性能损失，特征去除将停止。

包裹法将特征选择与分类器设计集成在一起，利用分类器进行特征选择。但包裹法计算量大，且随着学习器的改变最佳特征组合也会改变，有时会造成不利影响。

```
from sklearn.feature_selection import RFE
classifier=LogisticRegression(verbose=1)
rfe = RFE(estimator=classifier, n_features_to_select=200, step=0.05)
rfe.fit(x_train, y_train.astype('int'))
x_val_new=rfe.transform(x_val)
x_train_new=rfe.transform(x_train)
y_predict=classifier.predict(x_val_new)
total=0
right=0
for i in range(len(y_predict)):
    if y_predict[i]==y_val[i]:
        right+=1
    total+=1
acc=float(right/total)
print('Logistic Regression RFE val accuarcy: ' + str(acc))'''
```

(2)分类器设计与选择

和直接利用细胞原始数据分类相似，实验中对sklearn过滤法、相关系数法和PCA等3种降维方法与支持向量机、Logistic Regression、神经网络、随机森林4种分类方法进行了尝试，并通过5折交叉验证对各模型的参数进行选择，最终选择合适的降维方式与分类器对测试集标签进行预测。此部分代码详见`HMW2.py`.

由于要求说明哪些调控对于分类最有效，因而此时只能选用特征选择方法而不能再选用PCA等特征提取方法。但出于好奇心，实验中仍对PCA降维的方法进行了尝试。

a.支持向量机法

采用sklearn过滤法与支持向量机结合时，5折交叉验证的平均正确率为96.7%；采用相关系数法与支持向量机结合时，5折交叉验证的平均正确率为92.5%；采用PCA降维与支持向量机结合时，5折交叉验证的平均正确率为96.2%。

b.Logistic Regression

采用sklearn过滤法与Logistic Regression结合时，交叉验证的平均正确率为93.2%；采用相关系数法与Logistic Regression结合时，交叉验证平均正确率为90%；采用PCA降维与Logistic Regression结合时，交叉验证平均正确率为95.2%.

c.神经网络

采用sklearn过滤法与隐层节点数为100的单隐层神经网络结合时，平均正确率为94.9%；采用相关系数法进行特征选择时，神经网络的平均正确率为91.4%；采用PCA降维时平均正确率为96%.

d.随机森林方法

采用sklearn过滤法与随机森林结合时，5折交叉验证的平均正确率为91%；采用相关系数法选取特征时，随机森林的平均正确率为89%；采用PCA降维与随机森林结合时，交叉验证的平均正确率为85%。

(3)有效调控因子

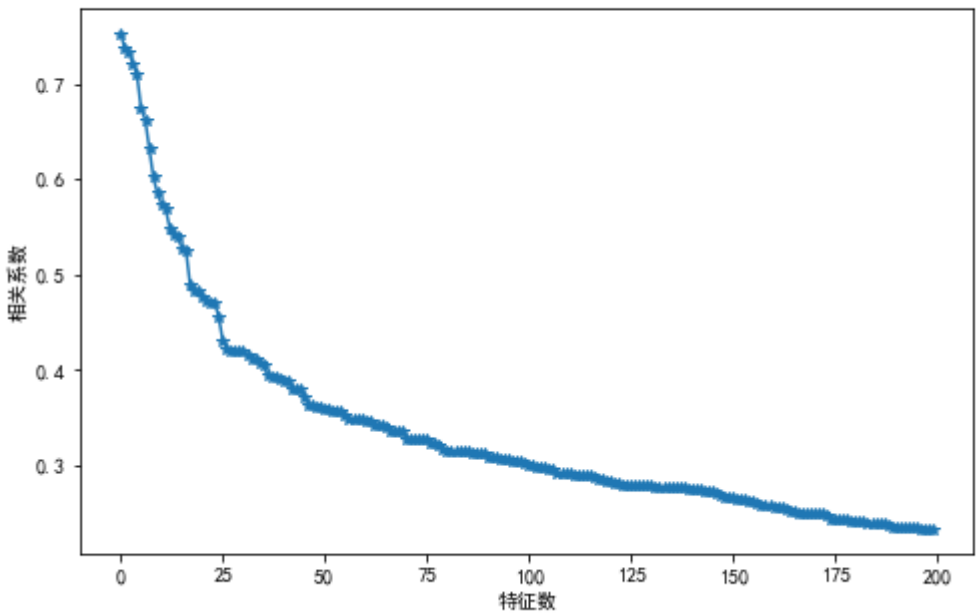
将利用sklearn过滤法与相关系数法挑选的前200维特征分别输出。设置sklearn过滤法所选总维数为200，则其所选择的特征维数为：

```
  2  20  37  41  47  55  66  68  70  74  87  93  96 102 104 106 109 111
 112 114 118 120 121 123 124 127 128 129 132 133 ......
```

相关系数法所选择的特征维数为：

```
127 106 139 154 151 325 111 121 343 335 339 320 333 321 109 334 133 323
 775 618 612 621 160 132 613 198 328 739 332 608 ......
```

绘制相关系数随特征数的变化，可以发现前25特征具有较大的相关系数，在说明前25维调控因子在分类中可能更为有效。



进一步探究所选特征数递增时验证集准确率变化，发现挑选5维特征时验证集准确率明显提高。对分类效果影响最大的5个调控因子为796,795,869,193,321.对应调控因子名称为ENSG00000187079_LINE3576_TEAD1_D_N3, ENSG00000007866_LINE3572_TEAD3_D_N1，ENSG00000112837_LINE19949_TBX18_I_N1, ENSG00000119725_LINE845_ZNF410_D, ENSG00000010030_LINE1801_ETV7_D, 可认为这些调控对于分类而言是最有效的。

(4)测试集预测

通过对各种特征筛选或降维以及分类的方法进行尝试，发现采用sklearn过滤法后模型分类效果更好且表现较为稳定。与直接利用细胞原始数据进行分类相似，采取类似于Adaboost算法的做法，每次随机选取80%的数据对采用过滤法进行特征选择后的数据分别采用不同的分类器进行训练与预测，每个模型均预测5次，最后将所有模型的预测结果进行投票，将出现次数最多的标签作为测试集最终的标签。

预测最终结果在`problem2.txt`中.

3. 降维、聚类和可视化

实验中共尝试了5种降维方法，每种降维方法均已在前两问中做过详细陈述，此处不再重复。此部分主要展示针对降维后数据的聚类与可视化。

根据先前实验结果选取最合适的降维方法后，分别采用PCA，t-SNE与LLE方法对降维后的训练数据进行二维可视化并利用KMeans对降维后的数据进行无监督聚类。通过将聚类结果和二维可视化图像进行对比，可以对聚类结果进行评价。

此部分代码分别包含在第1题与第2题的最后部分。

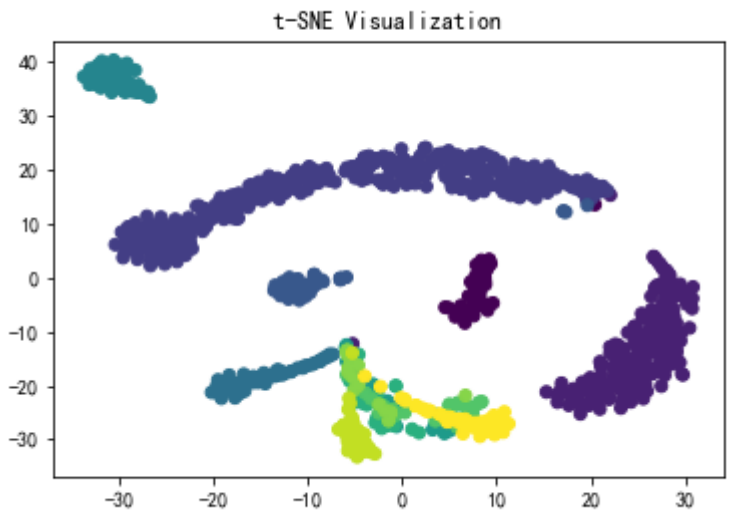
(1)细胞原始数据

a.降维与可视化

1）t-SNE利用概率分布来度量样本之间的距离（j样本作为i样本近邻的条件概率）

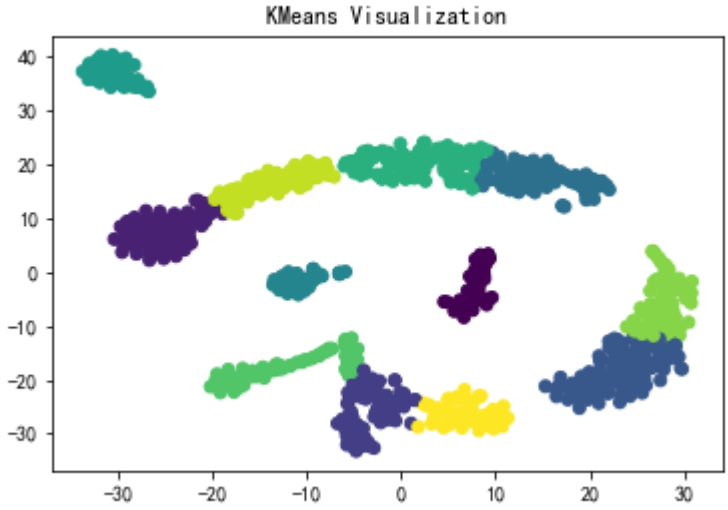
- 计算高维空间的分布$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)}$
- 降维后的分布$q_{ij} = \frac{(1+||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1+||y_k - y_l||^2)^{-1}}$
- 计算分布之间的差异程度KL距离，利用梯度下降等方法优化损失函数

t-SNE嵌入空间只能是2维或3维，对原始数据进行特征筛选后采用t-SNE得到二维可视化如下：



可以看出可视化效果较为理想。除个别类别外，各类别可以较为明显地分开。

利用Kmeans聚类如下：

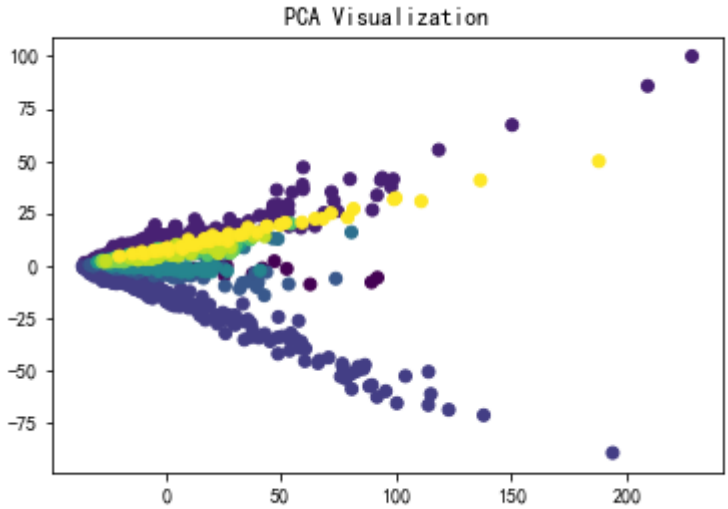


可以看出聚类后有些类别上产生了较大的偏差，如原先在二维可视化中横跨两个区域的浅紫色在聚类后被分为了四个类别；处于同一区域的深紫色则被分为了两个小类；原先有明显分类面的孤立的类别得到了较好的保留。

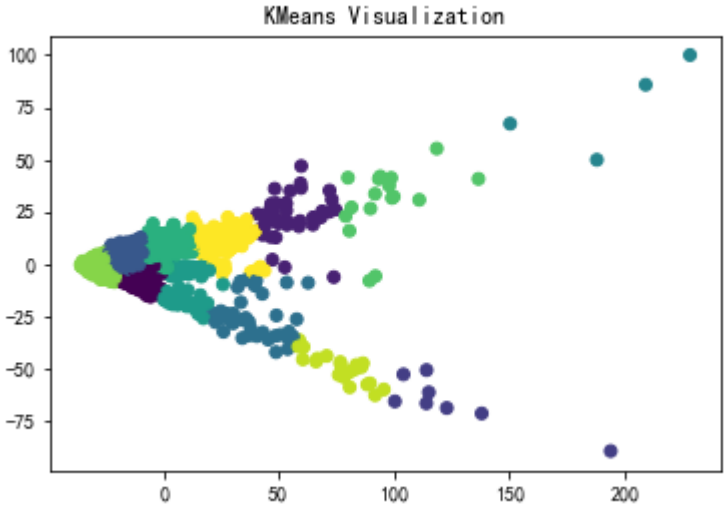
2）采用PCA降维方法时，算法流程如下：

- 计算样本均值
- 对数据进行中心化
- 计算协方差矩阵
- 对协方差矩阵进行特征值分解
- 取最大的k个特征值所对应的特征向量构成投影矩阵W
- 投影到$X^{'} = W^{T}X$

  特征筛选与降维后数据二维可视化如下:



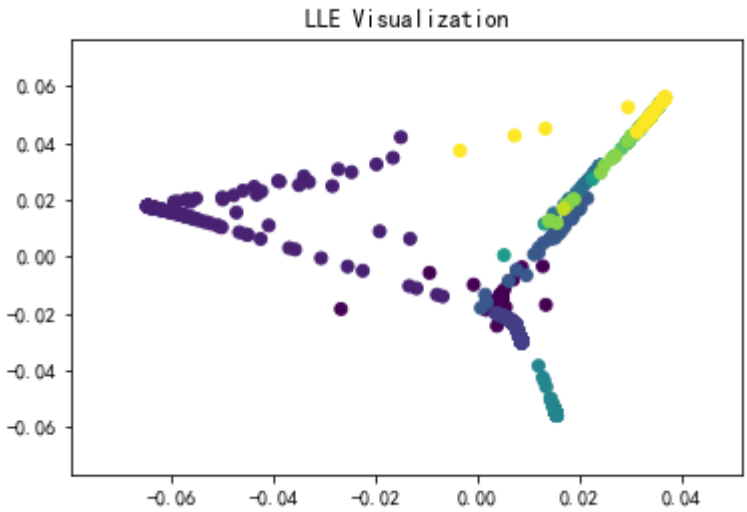
利用Kmeans聚类如下：



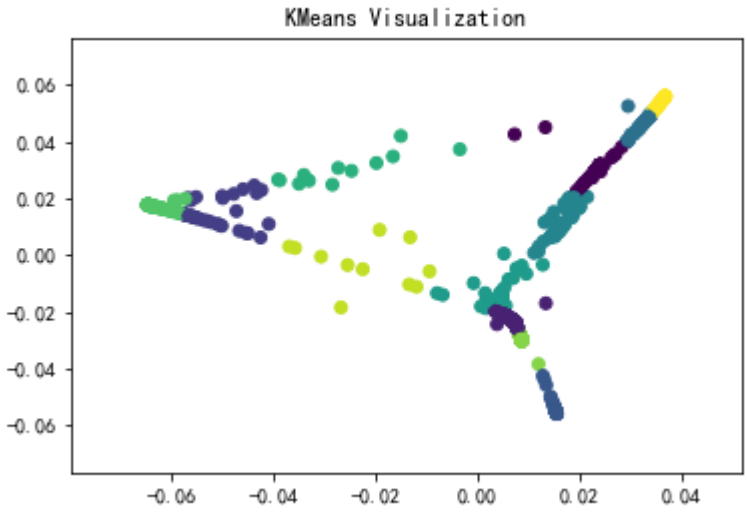
可以发现聚类结果与实际二维可视化存在一定差距。二维可视化后同类数据呈放射状线性分布，而聚类时则变为了块状分布，产生了较大的误差。

3）采用LLE降维方法时，算法流程如下：

- 原空间选择近邻，得到由近邻重构X误差最小的权重W_{ij}，$\epsilon(W)=\sum_i|X_i-\sum_j W_{ij}X_j|^2$
- 利用W_{ij}在低维空间得到重构误差最小的Y，$\Phi(Y)=\sum_i|Y_i-\sum_j W_{ij}Y_j|^2$



利用Kmeans聚类如下：

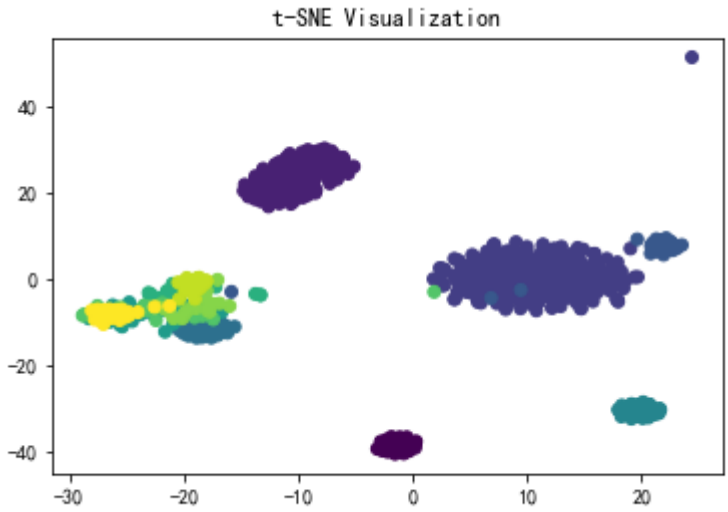


4）可视化效果对比分析

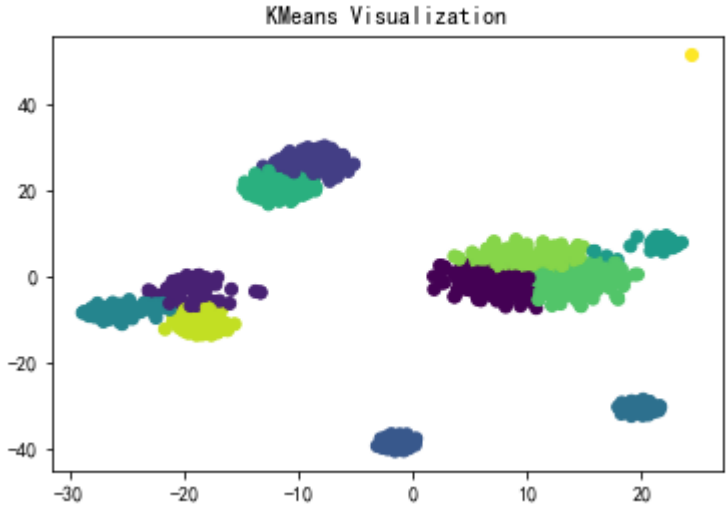
PCA属于线性降维算法，t-SNE与LLE属于非线性降维算法。PCA计算方法简单，降维可视化后各类数据间存在一定程度的重合，没有明显的分界面。t-SNE能有效将数据投影到低维空间，降维映射后数据在低维空间中仍能被分开，但t-SNE计算复杂度大，计算耗时长。LLE可视化后数据在低维空间中则接近线性分布较有特点。

(2)调控因子数据

1）采用t-SNE降维方法时，二维可视化如下：

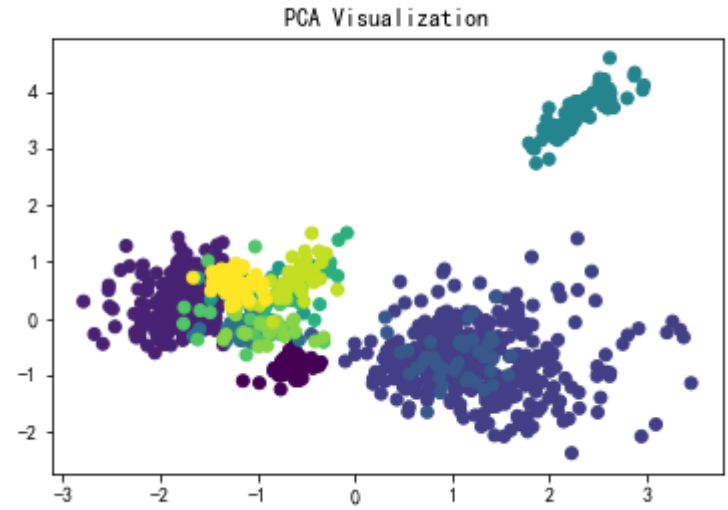


利用Kmeans聚类如下：

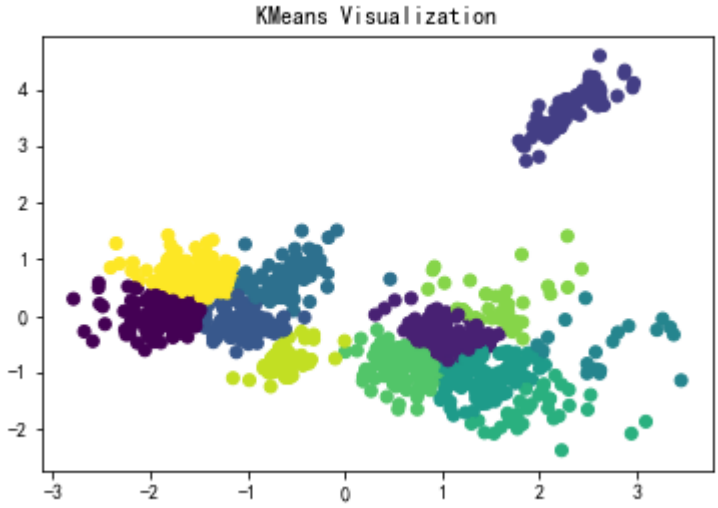


可以看出聚类结果存在一定偏差。和利用细胞原始数据可视化时的情况相近，聚类后出现了整块区域被划分为多个类别的现象，不过原先孤立的类别得到了较好的保留。

2）采用PCA降维方法时，二维可视化如下：

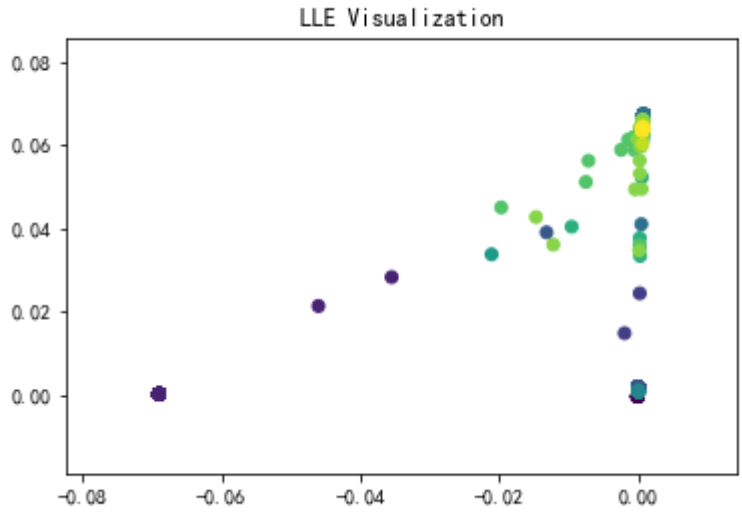


利用Kmeans聚类如下：

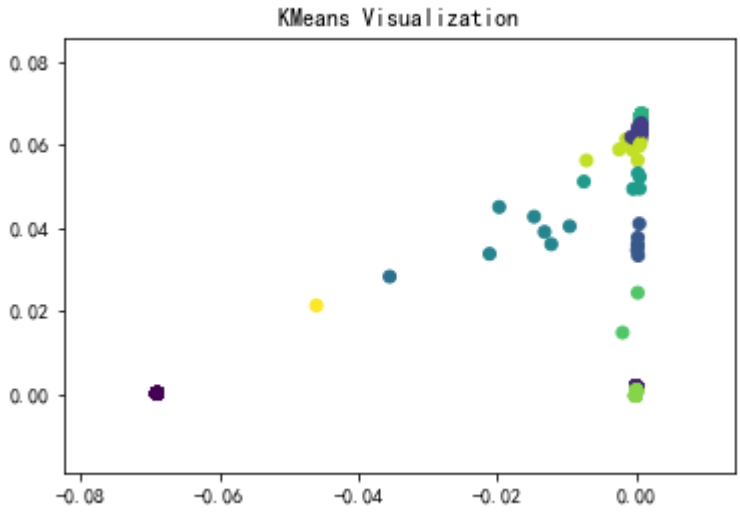


此时聚类结果与实际二维可视化结果差距较大。

3）采用LLE降维方法时，二维可视化如下：



利用Kmeans聚类如下：



个别类存在重合外，许多类别已成为分立的点可以直接分开，可视化效果得到提高。

4.实验总结

模式识别是这学期我所上的任务量最大、要求最严格的一门课，也是这学期令我收获最大的一门课。老师对待课程十分认真，一个学期的教学内容几乎涵盖了所有主流的机器学习与部分统计学习的方法；助教们也非常负责，不管是对课程内容还是课程作业有疑惑，助教们总会提供最及时的解答。

每两周一次的课程作业和以准确率排名为给分标准的大作业曾一度带给我很大压力，但正是一次次的实践与应用使我对各个算法的特点与适用情景有了更深的理解，也使我的编程能力得到了很大提高。

衷心祝愿模式识别课程能够越办越好！