作业2：线性分类模型

1. Fisher准则的最小二乘法推导

(1) 证明最优化如下 $\omega_0 = -\omega^T m$，其中 m为所有样本的均值

证明：

$$E = \frac{1}{2}\sum_{i=1}^{n}(\omega^T x_i + \omega_0 - t_i)^2$$

$$\frac{\partial E}{\partial \omega_0} = \sum_{i=1}^{n}(\omega^T x_i + \omega_0 - t_i) = 0$$

$$\omega_0 = \frac{1}{n}\sum_{i=1}^{n} t_i - \frac{1}{n}\sum_{i=1}^{n}\omega^T x_i = \frac{1}{n}\sum_{i=1}^{n} t_i - \omega^T m$$

由于

$$\sum_{i=1}^{n} t_i = \frac{n}{n_1}n_1 - \frac{n}{n_2}n_2 = 0$$

故

$$\omega_0 = -\omega^T m$$

(2) 推导最优化ω 服从下面的等式：

$$(S_\omega + \frac{n_1 n_2}{n} S_B)\ \omega = n(m_1 - m_2)$$

证明：

$$E = \frac{1}{2}\sum_{i=1}^{n}(\omega^T x_i + \omega_0 - t_i)^2$$

$$\frac{\partial E}{\partial \omega} = \sum_{i=1}^{n}(\omega^T x_i + \omega_0 - t_i)x_i = \sum_{i=1}^{n}(\omega^T x_i - \omega^T m - t_i)x_i = 0$$

由于

$$\sum_{i=1}^{n} x_i = nm\ ,\quad \sum_{i\in C_1} x_i = n_1 m_1\ ,\quad \sum_{i\in C_2} x_i = n_2 m_2$$

故有：

$$\Rightarrow \sum_{i=1}^{n} x_i x_i^T \omega - nmm^T\omega - \sum_{i\in C_1}\frac{n}{n_1}x_i + \sum_{i\in C_2}\frac{n}{n_2}x_i = 0$$

$$\Rightarrow \sum_{i=1}^{n} x_i x_i^T \omega - nmm^T\omega = n(m_1 - m_2)$$

由于

$$S_\omega = \sum_{i\in C_1}(x_i - m_1)(x_i - m_1)^T + \sum_{i\in C_2}(x_i - m_2)(x_i - m_2)^T$$

$$= \sum_{i=1}^{n} x_i x_i^T - 2m_1\sum_{i\in C_1} x_i^T + n_1 m_1 m_1^T - 2m_2\sum_{i\in C_2} x_i^T + n_2 m_2 m_2^T$$

$$= \sum_{i=1}^{n} x_i x_i^T - n_1 m_1 m_1^T - n_2 m_2 m_2^T$$

$$S_B = (m_2 - m_1)(m_2 - m_1)^T = m_2 m_2^T - m_1 m_2^T - m_2 m_1^T + m_1 m_1^T$$

故:

$$S_\omega + \frac{n_1 n_2}{n} S_B = \sum_{i=1}^{n} x_i x_i^T - n_1 m_1 m_1^T - n_2 m_2 m_2^T + \frac{n_1 n_2}{n}(m_2 m_2^T - m_1 m_2^T - m_2 m_1^T + m_1 m_1^T)$$

$$= \sum_{i=1}^{n} x_i x_i^T - (\frac{n_1^2}{n} m_1 m_1^T + \frac{n_2^2}{n} m_2 m_2^T + \frac{n_1 n_2}{n} m_1 m_2^T + \frac{n_1 n_2}{n} m_2 m_1^T)$$

$$= \sum_{i=1}^{n} x_i x_i^T - \frac{1}{n}(n_1 m_1 + n_2 m_2)(n_1 m_1 + n_2 m_2)^T$$

$$= \sum_{i=1}^{n} x_i x_i^T - nmm^T$$

因此:

$$(S_\omega + \frac{n_1 n_2}{n} S_B)\omega = \sum_{i=1}^{n} x_i x_i^T \omega - nmm^T\omega = n(m_1 - m_2)$$

原结论得证

(3) 通过（2）推导出$\omega \propto S_\omega^{-1}(m_1 - m_2)$，这意味着我们得到了与Fisher准则相同的形式

由于$S_B = (m_2 - m_1)(m_2 - m_1)^T$),对$\forall\omega$

$$S_B\omega = (m_2 - m_1)(m_2 - m_1)^T\omega = (m_2 - m_1)[(m_2 - m_1)^T\omega]$$

其中$[(m_2 - m_1)^T\omega]$为标量，故$S_B\omega = k(m_2 - m_1)$

故

$$
\begin{aligned}
&(S_\omega + \frac{n_1 n_2}{n} S_B)\omega = n(m_1 - m_2) \\
\Rightarrow\ & S_\omega \omega + \frac{n_1 n_2}{n} k(m_2 - m_1) = n(m_1 - m_2) \\
\Rightarrow\ & S_\omega \omega = (n + \frac{n_1 n_2}{n} k)(m_1 - m_2) \\
\Rightarrow\ & \omega \propto S_\omega^{-1}(m_1 - m_2)
\end{aligned}
$$

∴原结论得证

2. 二分类问题：乳腺癌数据集分类

2.0 数据预处理

运行二分类问题程序`HM2_2_1.py`与`HM2_2_2.py`时请将数据集`breast-cancer-wisconsin.txt`放在与脚本相同的当前目录下，以便程序进行数据的读取。

由于数据中存在缺失值，首先利用pandas读取数据，用NaN替换缺失值"?"，以便进一步处理

```
df=pd.read_csv('breast-cancer-wisconsin.txt',delimiter='\t',header=None)
df.replace(to_replace='?',value=np.nan,inplace=True)
for u in df.columns:
  df[u]=pd.to_numeric(df[u])
```

对于logistic回归，共尝试使用了4种处理缺失值方法，包含剔除含有缺失值的样本，用均值替代缺省值，用中值替代缺省值，与用众数替代缺省值等。

```
df=df.dropna()#用舍去的方法处理缺省值

imp=Imputer(missing_values='NaN',strategy='mean',axis=0)#用平均值替代缺省值
x_test=imp.fit_transform(x_test)
x_train=imp.fit_transform(x_train)

imp=Imputer(missing_values='NaN',strategy='median',axis=0)#用中位数替代缺省值
imp=Imputer(missing_values='NaN',strategy='most_frequent',axis=0)#用众数替代缺省值
```

随机划分75%数据作为训练集，25%数据作为测试集：

```
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test =
train_test_split(new_df[:,1:10],new_df[:,10],test_size=0.25)
```

2.1 Logistic 回归

采用sklearn自带的LogisticRegression()函数实现分类

```
classifier=LogisticRegression()
classifier.fit(x_train,y_train)
```

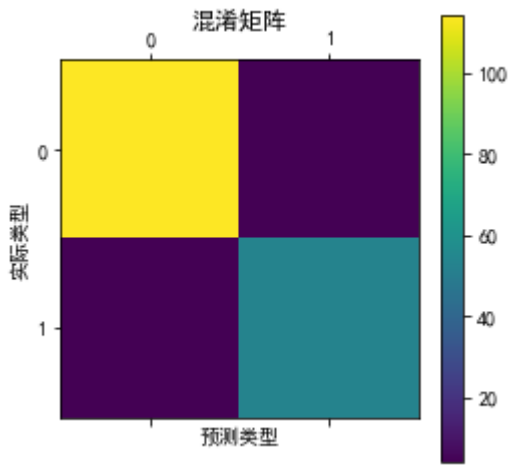
```
y_predict=classifier.predict(x_test)

sum=0
scores_list=[]
for j in range(5):
  for i in range(y_predict.shape[0]):
    if y_predict[i]==y_test[i]:
        sum=sum+1
  score=sum/y_predict.shape[0]
  sum=0
  scores_list.append(score)

scores=np.array(scores_list)
print("accuracy",np.mean(scores),scores)
```

其中准确率结果为在测试集上5次测试结果的平均值，采用不同方法处理缺失值时准确率如下，可以看出Logistic方法测试集准确性在96%左右

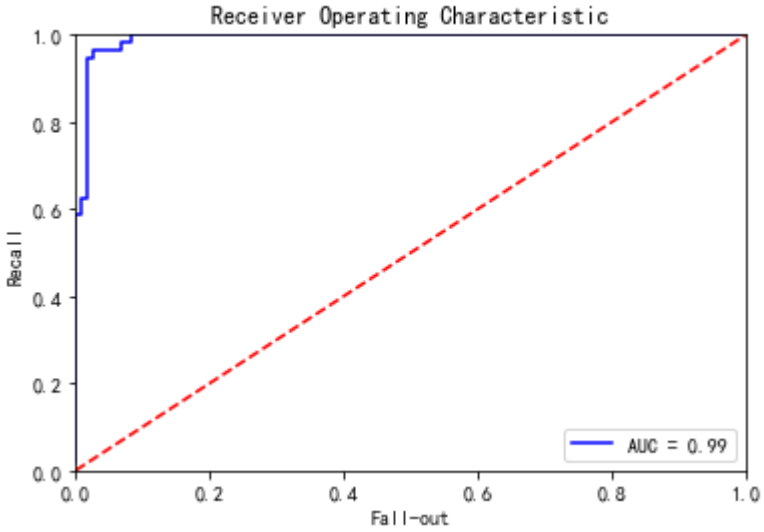
```
#忽略缺省值
accuracy 0.96076544936 [ 0.92105263  0.94444444  1.          0.96774194  0.97058824]
##用中位数替代缺省值
accuracy 0.961014957265 [ 1.          0.96875     0.8974359   0.97222222  0.96666667]
#用平均数替代缺省值
accuracy 0.961807081807 [ 0.94871795  0.97222222  1.          0.97142857  0.91666667]
#用众数替代缺省值
accuracy 0.951920341394 [ 0.97297297  0.94594595  1.          0.89473684  0.94594595]
```

采用混淆矩阵与ROC曲线进一步评估分类效果，以平均数替代缺省值，得到测试集上的混淆矩阵与ROC曲线如下：

```
confusion_matrix=confusion_matrix(y_test,y_predict)
print (confusion_matrix)
```




```
#混淆矩阵
[[114   4]
 [  4  53]]
```



2.2 Fisher 线性判别

利用第1题与Fisher线性判别的公式求取权向量ω如下:

```
m_0=np.mean(x_0,axis=0)#计算每一列的均值
m_1=np.mean(x_1,axis=0)
n_0=x_0.shape[0]#计算每一类的数量
n_1=x_1.shape[0]
cov_0=np.cov(x_0.T)#计算每一类的协方差矩阵
cov_1=np.cov(x_1.T)
Sw= ((n_0-1)*cov_0+ (n_1-1)*cov_1)
Sw_1=np.linalg.inv(np.mat(Sw))
w=Sw_1*np.mat(m_0-m_1).T*(n_0+n_1)#得到权向量
```

权向量ω决定超平面方向，阈值权ω_0决定它的位置。当样本为正态分布且两类协方差相同时，如果把样本的算术平均作为均值估计，样本的协方差矩阵作为真实协方差矩阵的估计，Fisher线性判别所得方向与最优贝叶斯决策方向相同，则ω_0可表示为:

$$\omega_0 = -\frac{1}{2}(m_1 + m_2)^T S_\omega^{-1}(m_1 - m_2) - ln\frac{P(\omega_2)}{P(\omega_1)}$$

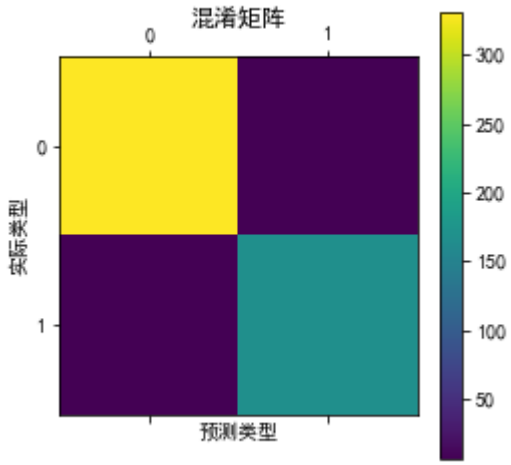
不考虑先验时，则可采用阈值:

$$\omega_0 = -\frac{1}{2}(\widetilde{m_1} + \widetilde{m_2})$$

此处选用第2种公式.

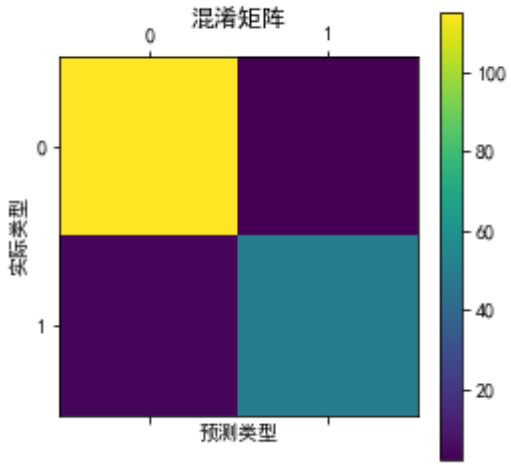
以忽略缺省值的方式对数据进行预处理，得到训练集的准确率与混淆矩阵如下:

```
#训练集准确率
accuracy 0.966796875 [ 0.96679688  0.96679688  0.96679688  0.96679688  0.96679688]
#混淆矩阵
[[331   6]
 [  7 168]]
```



测试集准确率与混淆矩阵如下：

```
#测试集准确率:
accuracy 0.964912280702 [ 0.96491228  0.96491228  0.96491228  0.96491228  0.96491228]
#混淆矩阵
[[115   2]
 [  4  50]]
```



Fisher 线性判据正确率约为 96.5%. 可以看到 logistic 回归和 Fisher 线性回归得到了相似的结果，分类结果较好。

3. Logistic Regression的多类推广：Softmax Regression

3.1 分类器类别

Softmax Regression是线性分类器，因为分界面在特征空间是线性的。

线性判别函数的一般形式为$g(x)=\omega^T x+\omega_0$， $g(x)=0$定义决策面。$g(x)$是线性函数时，决策面是超平面。令$g(x)=g_1(x)-g_2(x)$，定义决策规则：$g(x)>0$，则决策$x\in\omega_1$；$g(x)<0$，则决策$x\in\omega_2$。

对于logistic模型：

$$logit(x) = ln(\frac{P(y \mid x)}{1 - P(y \mid x)}) = \alpha + \beta x$$

可见logistic回归的决策函数具有线性分类器的特征，其决策规则如下：$logit(x) > 0$，则决策$x \in \omega_1$；$logit(x) < 0$，则决策$x \in \omega_2$。

Softmax回归模型是logistic回归模型在多分类问题上的推广，推广到多元，样本属于y=1类的概率：

$$P(y \mid x) = \frac{e^{\beta_0 + \beta_1 x_1 + ... + \beta_m x_m}}{1 + e^{\beta_0 + \beta_1 x_1 + ... + \beta_m x_m}}$$

则多元的logit函数是

$$logit(x) = ln(\frac{P(y \mid x)}{1 - P(y \mid x)}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m$$

可以看出分界面上的点是线性的，Softmax Regression是线性分类器。

3.2 图像分类

多元逻辑回归常见方法有one-vs-rest(OvR)和many-vs-many(MvM)两种。OvR将多元逻辑回归问题看做二元逻辑回归问题处理，对于第K类的分类决策，把所有第K类的样本作为正例，除了第K类样本外的所有样本作为负例进行二元逻辑回归，得到第K类的分类模型； MvM则需要对多类中的每两类构造分类器，如果模型有T类，则每次需要在所有T类样本里面选择两类样本T1类和T2类，将所有输出为T1和T2的样本进行二元逻辑回归，以T1作为正例，T2作为负例得到模型参数，对于T类样本MvM共需要T(T-1)/2次分类。

由于逐对分类不会出现两类样本数过于不均衡的问题且决策歧义的区域更小，一般而言MvM分类相对精确。使用sklearn中的LogisticRegression函数，通过设置参数multi_class可实现多分类问题。此处选用lbfgs作为优化损失函数，使用LogisticRegression函数进行Softmax回归。

运行图像分类代码`HM3_2.py`时请将按人像分类的含有照片的各个小文件夹（如0,1，...）放在与脚本相同的目录下，以便程序进行数据的读取。

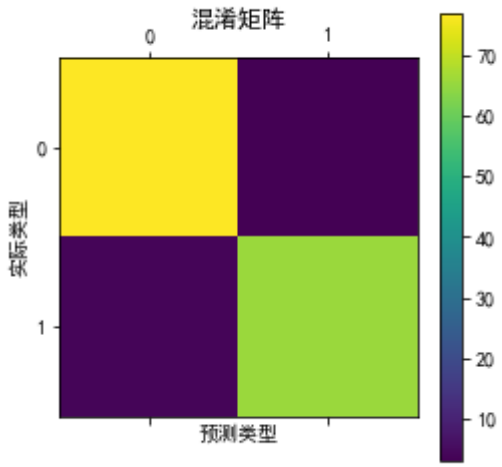
进行数据预处理时，将序列[0,1,2,...,9]进行随机排序，取前n个作为将要用于分类的类别。将选中类别的文件夹内的所有图片读出，以数组形式储存所有[样本，标签]，构成数据集。训练集与测试集划分与第2题相同。

```
iRandom=[]
random_list=list(range(0,10))
random.shuffle(random_list)
iRandom=random_list[0:2]

packet_list=[]
for num in iRandom:
    line='./'+str(num)+'/'
    packet_list.append(line)
data_list=[]

for i in range(len(packet_list)):
    for filename in os.listdir(packet_list[i]):
        img_name=packet_list[i]+filename
        img = io.imread(img_name, as_grey=True)
```

```
        arr=img.flatten()
        arr_list=arr.tolist()
        arr_list.append(iRandom[i])
        data_list.append(arr_list)
data=np.array(data_list)
```

（1）请随机取出2个人的图像，75%作为训练集，25%作为测试集，给出Softmax的测试集正确率。

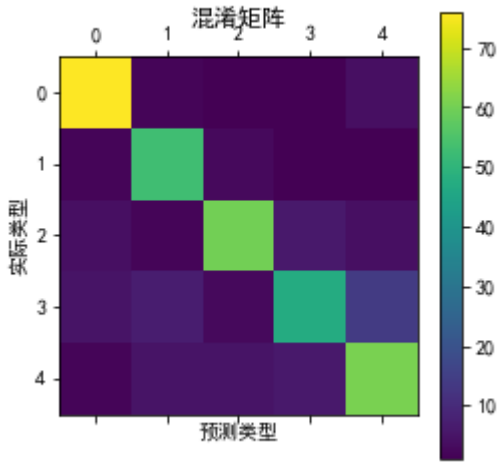
```
#随机选取种类
[9, 1]
accuracy:  0.953333333333
             precision    recall  f1-score   support

        1.0       0.95      0.96      0.96        80
        9.0       0.96      0.94      0.95        70
avg / total       0.95      0.95      0.95       150
#混淆矩阵
[[77  3]
 [ 4 66]]
```



（2）随机取出5个人的图像，75%作为训练集，25%作为测试集，给出Softmax的测试集正确率

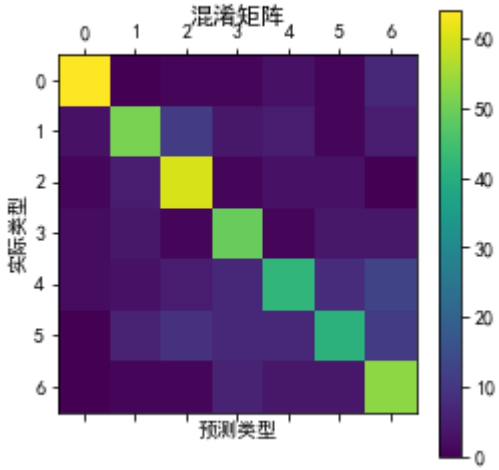
```
#随机选取种类
[5, 0, 4, 9, 6]
accuracy:  0.792
             precision    recall  f1-score   support

        0.0       0.85      0.90      0.88        84
        4.0       0.77      0.88      0.82        60
        5.0       0.83      0.79      0.81        76
        6.0       0.77      0.62      0.69        76
        9.0       0.73      0.77      0.75        79
avg / total       0.79      0.79      0.79       375
#混淆矩阵
[[76  2  1  1  4]
 [ 2 53  3  1  1]
```

```
 [ 4  2 60  6  4]
 [ 5  7  3 47 14]
 [ 2  5  5  6 61]]
```



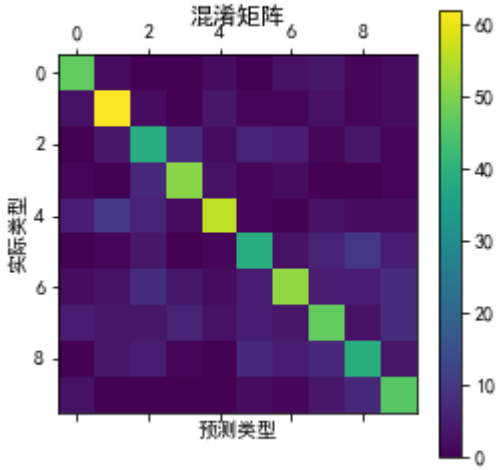
(3)随机取出7个人的图像，75%作为训练集，25%作为测试集，给出Softmax的测试集正确率。

```
#随机选取种类
[2, 6, 5, 0, 9, 7, 3]
accuracy:  0.685714285714
             precision    recall  f1-score   support

        0.0       0.89      0.83      0.86        77
        2.0       0.73      0.64      0.68        80
        3.0       0.68      0.82      0.75        73
        5.0       0.65      0.75      0.70        65
        6.0       0.65      0.53      0.58        80
        7.0       0.66      0.51      0.57        81
        9.0       0.57      0.77      0.65        69

avg / total       0.69      0.69      0.68       525
#混淆矩阵
[[64  0  1  1  3  1  7]
 [ 3 51 11  4  5  1  5]
 [ 1  5 60  1  3  3  0]
 [ 2  4  1 49  1  4  4]
 [ 2  3  5  7 42  8 13]
 [ 0  6  9  7  7 41 11]
 [ 0  1  1  6  4  4 53]]
```



(4)使用所有人的图像，75%作为训练集，25%作为测试集，给出Softmax的测试集正确率

```
#随机选取种类
[2, 4, 7, 6, 1, 0, 3, 8, 9, 5]
accuracy:  0.637333333333
             precision    recall  f1-score   support

        0.0       0.71      0.77      0.74        61
        1.0       0.69      0.78      0.73        79
        2.0       0.52      0.56      0.54        70
        3.0       0.71      0.77      0.74        66
        4.0       0.77      0.64      0.70        87
        5.0       0.58      0.57      0.57        69
        6.0       0.68      0.55      0.61        94
        7.0       0.59      0.53      0.56        89
        8.0       0.54      0.54      0.54        72
        9.0       0.58      0.73      0.65        63

avg / total       0.64      0.64      0.64       750
#混淆矩阵
[[47  2  0  0  2  0  3  4  1  2]
 [ 3 62  2  0  4  1  1  3  1  2]
 [ 0  4 39  8  2  6  5  1  4  1]
 [ 1  0  7 51  3  1  2  0  0  1]
 [ 5 10  6  2 56  1  0  3  2  2]
 [ 0  1  4  0  1 39  3  6 10  5]
 [ 2  3  8  4  2  5 52  5  5  8]
 [ 5  4  4  6  3  5  4 47  3  8]
 [ 0  4  5  1  0  7  5  7 39  4]
 [ 3  0  0  0  0  2  1  4  7 46]]
```



探究分类正确率随分类种类的变化趋势时，随机选取的种类数依次递增时得到平均的统计结果大致如下

Class count	**1**	**2**	**3**	**4**	**5**
Accuracy	100%	92%	85%	75%	71%
Class count	6	7	8	9	10
Accuracy	69%	69%	66%	65%	64%

观察实验可以发现，分类数为2时准确率的变化范围较大；随着种类数的增多，准确率浮动逐渐减小，且分类正确率明显下降。

对于二分类问题，不同类别间的特征差异在随机选取时有明显区别，有些类别较易分类（如第0类与第1类），有些类别则不易区分（如第6类与第9类）。对特征差异较明显的两类进行分类时将明显取得较好的分类效果。

当分类种类逐渐变多时，由混淆矩阵可以很明显地看出被分错的测试样本数增多了。容易理解的是，样本种类越多，含有重叠或相近特征的可能性越大，歧义区域越多，越容易产生分类错误。因而随着随机选取的分类种类的增多，Softmax Regression的正确率会下降。