5241 HW2
hl3115
Haiqi Li

Problem 1

It's obvious that $E(R_{te}(\hat{\beta})) \leq E(R_{te}(\hat{\beta}))$ since $\hat{\beta}$ is least square estimator of test data ~~de~~ set.

Also $\sum_{i=1}^{N} E(y_i - \hat{\beta}^T x_i)^2 = N E(R_{tr}(\hat{\beta})) = (N-p-1)\sigma^2$ $\quad \sum_{i=1}^{M} E(\tilde{y}_i - \hat{\beta} \tilde{x}_i)^2 = M E(R_{te}(\hat{\beta})) = (M-p-1)\sigma^2$

if $M=N$ then $E(R_{tr}(\hat{\beta})) = E(R_{te}(\hat{\beta}))$ then we can ~~do~~ conclude

$$E(R_{tr}(\hat{\beta})) = E(R_{te}(\hat{\beta})) \leq E(R_{te}(\hat{\beta}))$$

if $M < N$

suppose N is not changed. Then if $E(R_{te}(\hat{\beta}))$ is not ~~chg.~~ changed due to ~~the~~ decreasing of M, then the proof is over.

$$E(R_{te}(\hat{\beta})) \underset{①}{=} E\left(\frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i - \hat{\beta}^T \tilde{x}_i)^2\right) \underset{②}{=} E\left(E\left(\frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i - \hat{\beta}^T\tilde{x}_i)^2 \mid \hat{\beta}\right)\right) \quad (E(X) = E(E(X|Y)))$$

① is solving expectation of $\hat{\beta}$, i.e. $E(f(\hat{\beta}))$ ② is like $E(E(g(\tilde{x}_i \tilde{y}_i) \mid \hat{\beta})))$ the first E try to compute expectation of $\hat{\beta}$, the second "E" in first "E" is that given $\hat{\beta}$, we try to solve expectation of $\tilde{x}_i\, \tilde{y}_i$, or to say $E_{\hat{\beta}}\left(E_{\tilde{x}_i \tilde{y}_i}\left(\frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i - \hat{\beta}^T\tilde{x}_i)^2 \mid \hat{\beta}\right)\right)$

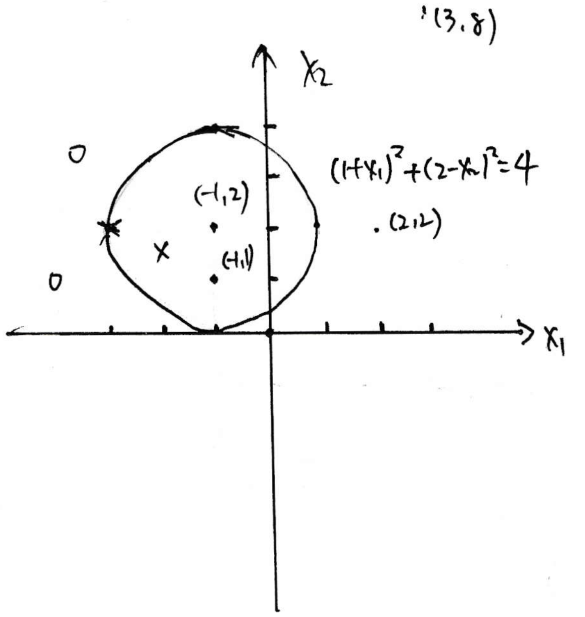
Since $\tilde{x}_i, \tilde{y}_i$ are iid then $E_{\hat{\beta}}\left(E_{\tilde{x}_i\tilde{y}_i}\left(\frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i - \hat{\beta}^T\tilde{x}_i)^2 \mid \hat{\beta}\right)\right) = E_{\hat{\beta}}\left(E_{\tilde{x}_i\tilde{y}_i}\left((\tilde{y}_i - \hat{\beta}^T\tilde{x}_i)^2 \mid \hat{\beta}\right)\right)$
$= E(\tilde{y}_i - \hat{\beta}^T\tilde{x}_i)$

So we conclude that $E(R_{te}(\hat{\beta})) = E(\tilde{y}_i - \hat{\beta}^T\tilde{x}_i)$ has nothing to do with M.

So $E(R_{tr}(\hat{\beta})) < E(R_{te}(\hat{\beta}))$ in $M<N$ //

Problem 2

(a)



(b) $(1+x_1)^2+(2-x_2)^2>4$ are points out of circle. (e.x. points in o)

$(1+x_1)^2+(2-x_2)^2\leq 4$ are points on and in the circle. (e.x. points in x)

(c)

point	position	color
(0,0)	out	blue
(-1,1)	in	red
(2,2)	out	blue
(3,8)	out	blue.

(d) $(1+x_1)^2+(2-x_2)^2>4$

$\Rightarrow 5+2x_1-4x_2+x_1^2+x_2^2>4$

I think it is a linear in terms of x_1, x_1^2, x_2, x_2^2.

5241 HW2 Haiqi Li hl3115

February 18, 2018

1 Problem 3

In [1]:
```
import numpy as np
import pandas as pd
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.preprocessing import scale
from sklearn.linear_model import LogisticRegression
from sklearn.decomposition import IncrementalPCA
```

In [2]:
```
train3=pd.read_table(r'C:\Users\shanaxmiku\Desktop\files\train_3.txt',sep=",",header=No
train5=pd.read_table(r'C:\Users\shanaxmiku\Desktop\files\train_5.txt',sep=",",header=No
train8=pd.read_table(r'C:\Users\shanaxmiku\Desktop\files\train_8.txt',sep=",",header=No
test_raw=pd.read_table(r'C:\Users\shanaxmiku\Desktop\files\zip_test.txt',sep=" ",header
n3=train3.shape[0]
n5=train5.shape[0]
n8=train8.shape[0]
nwhole=n3+n5+n8
_=np.array(["3","5","8"])
label=np.repeat(_,[n3,n5,n8],axis=0)
data=pd.concat([train3,train5,train8])
data=np.matrix(data)

#dealing with test data
test_raw=np.matrix(test_raw)
test_label_raw=test_raw[:,0]
goodvalues=[3,5,8]
draw=np.where(test_label_raw==goodvalues)
test_data=test_raw[:,1:]
test_label=test_label_raw[draw[0]]
test_label=np.array(test_label.T)[0]
test_label=test_label.astype('int').astype('str')
test_data=test_data[draw[0],:]
```

Get data an make a raw transformation.This also includes the part of searching 3,5 and 8 in test data.

1.1 Question 1

```
In [3]: clf=LDA()
        clf.fit(data,label.ravel())

        train_predict=clf.predict(data)
        train_error=1-np.mean(train_predict==label)
        print("Training data error: %f"%train_error)

        test_predict=clf.predict(test_data)
        test_error=1-np.mean(test_predict==test_label)
        print("Test data error: %f"%test_error)
```

```
Training data error: 0.015945
Test data error: 0.087398
```

1.2 Question 2

```
In [4]: pca= IncrementalPCA(n_components=49)
        pca.fit(data,label)
        U=pca.transform(data)
        clf_lda=LDA()
        train_pca=U
        clf_lda.fit(train_pca,label.ravel())

        train_pca_predict=clf_lda.predict(train_pca)
        train_pca_error=1-np.mean(train_pca_predict==label)
        print("Training data error after PCA: %f"%train_pca_error)

        U=pca.transform(test_data)
        test_pca=U
        test_pca_predict=clf_lda.predict(test_pca)
        test_pca_error=1-np.mean(test_pca_predict==test_label)
        print("Test data error after PCA: %f"%test_pca_error)
```

```
Training data error after PCA: 0.042141
Test data error after PCA: 0.087398
```

1.3 Question 3

```
In [5]: temp1=np.kron(np.identity(8),np.array([0.5,0.5]).reshape(2,1))
        filter=np.kron(temp1,temp1)
        #kronecker product learned in deep learning CNN.

        data_filted=np.dot(data,filter)
        test_filted=np.dot(test_data,filter)
        clf_filt=LDA()
```

```
clf_filt.fit(data_filted,label)
train_filt_predict=clf_filt.predict(data_filted)
train_filt_error=1-np.mean(train_filt_predict==label)
print("Training filted data error: %f"%train_filt_error)

test_filt_predict=clf_filt.predict(test_filted)
test_filt_error=1-np.mean(test_filt_predict==test_label)
print("Test filted data error: %f"%test_filt_error)
```

```
Training filted data error: 0.033599
Test filted data error: 0.075203
```

```
In [6]: clf_multinominal=LogisticRegression(multi_class='multinomial',solver='lbfgs')
        clf_multinominal.fit(data_filted,label)
        train_mul_predict=clf_multinominal.predict(data_filted)
        train_mul_error=1-np.mean(train_mul_predict==label)
        print("Training filted data error under multinomial: %f"%train_mul_error)
        test_mul_predict=clf_multinominal.predict(test_filted)
        test_mul_error=1-np.mean(test_mul_predict==test_label)
        print("Test filted data error under nultinomial: %f"%test_mul_error)
```

```
Training filted data error under multinomial: 0.021640
Test filted data error under nultinomial: 0.085366
```

1.4 Summary

Training data error: 0.015945
Test data error: 0.087398
Training data error after PCA: 0.042141
Test data error after PCA: 0.087398
Training filted data error: 0.033599
Test filted data error: 0.075203
Training filted data error under multinomial: 0.021640
Test filted data error under nultinomial: 0.085366

As we can see,the raw LDA method has lowest training error which is reasonable since we are training on raw full data.

PCA lower the dimension and the actual information we use is decreased.But the performance after PCA is fairly good since the error rate of test data does not change too much.

Filted data is also a way to compress data.But this filter works since the test error is becoming lower.Actually, I know that this kind of filter is often a good way in dealing picture and vision issues.

Multinominal logistic regression (I learned as softmax method) performs good.The test error becoming lower compared with LDA method.