

In []:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Feb 28 18:26:25 2017

@author: huangshuhui
"""

###problem1###

##4.a##
from numpy import *
x=mat([[0,3,1],[1,3,1],[0,1,1],[1,1,1]])
y=mat([[1,1,0,0]])
w0=mat([[-2,1,0]])
product=x*w0.T
s0=[]
lambda0=0.07
for i in range(0,4):
    s=1/(1+e**int(-product[i]))
    s0.append(s)

##4.b##
Omega=diag(s0)-diag(s0)*diag(s0)
e0=(x.T*Omega*x+mat(diag([0.14,0.14,0.14]))).I*(x.T*((y-s0).T)-2*lambda0 *(w0.T)
)
w1=w0.T+e0

##4.c##
product1=x*w1
s1=[]
for i in range(0,4):
    s=1/(1+e**float(-product1[[i]]))
    s1.append(s)

##4.d##
Omega1=diag(s1)-diag(s1)*diag(s1)
e1=(x.T*Omega1*x+mat(diag([0.14,0.14,0.14]))).I*(x.T*((y-s1).T)-2*lambda0 *(w1))
w2=w1+e1

###problem4###
import scipy.io as sio
import matplotlib.pyplot as plt

wines=sio.loadmat('/Users/huangshuhui/Desktop/study/CS289/hw2017/hw4/data.mat')
wines_test=wines['X_test']
wines_train=wines['X']
wines_trainlab=wines['y']

##4.a##
w0=mat([0.0001]*12)
```

```

x=mat(wines_train)

y=mat(wines_trainlab)
testx=mat(wines_test)

lambd=0.05
def batchgra(nums,x,y,w0,lambd):
    w=w0
    x=x
    y=y
    costnew=[]
    i=0
    while i<nums:
        product=x*w.T
        s=[]
        for j in range(0,len(x)):
            s_mid=1/(1+e**float(-product[j]))
            s.append(s_mid)
        ei=x.T*(y-mat(s).T)
        cost=lambd*w*w.T-(y.T*mat(log(s)).T+(1-y).T*(log(1-mat(s)).T))
        w=(1-0.001/len(x)*2*lambd)*w+0.001/len(x)*ei.T
        costnew.append(float(cost))
        i=i+1
    return costnew,w

```

```

cost_batch,w_batch=batchgra(20,x,y,w0)

```

```

plt.plot(cost_batch,'b*')
plt.plot(cost_batch,'r')
plt.xlabel('times of iteration')
plt.ylabel('cost value')
plt.title('relation of iteration times and costs')

```

##4.b##

```

import random
def stochgra(nums,x,y,w0):
    w=w0
    x=x
    y=y
    costnew=[]
    i=0
    while i<nums:
        sample=random.sample(range(0,len(x)),1)
        producti=x[sample]*w.T
        product=x*w.T
        si=1/(1+e**float(-producti))
        ei=(y[sample]-si)*x[sample]
        s=[]
        for j in range(0,len(x)):
            s_mid=1/(1+e**float(-product[j]))
            s.append(s_mid)
        cost=0.05*w*w.T-(y.T*mat(log(s)).T+(1-y).T*(log(1-mat(s)).T))
        w=(1-0.0001)*w+0.001*ei
        costnew.append(float(cost))

```

```
i=i+1
```

```
return costnew,w
```

```
cost_sto,w_sto=stochgra(20,x,y,w0)
plt.plot(cost_sto,'b*')
plt.plot(cost_sto,'r')
plt.xlabel('times of iteration')
plt.ylabel('cost value')
plt.title('relation of iteration times and costs of stochastic gradient')
```

```
##4.c##
```

```
def stochgra2(nums,x,y,w0):
    w=w0
    x=x
    y=y
    costnew=[]
    i=0
    while i<nums:
        sample=random.sample(range(0,len(x)),1)
        producti=x[sample]*w.T
        product=x*w.T
        si=1/(1+e**float(-producti))
        ei=(y[sample]-si)*x[sample]
        s=[]
        for j in range(0,len(x)):
            s_mid=1/(1+e**float(-product[j]))
            s.append(s_mid)
        cost=0.05*w*w.T-(y.T*mat(log(s)).T+(1-y).T*(log(1-mat(s)).T))
        w=(1-0.0001/(1+i))*w+0.001/(1+i)*ei
        costnew.append(float(cost))
        i=i+1
    return costnew,w
```

```
cost_sto2,w_sto2=stochgra2(20,x,y,w0)
plt.plot(cost_sto2,'b*')
plt.plot(cost_sto2,'r')
plt.xlabel('times of iteration')
plt.ylabel('cost value')
plt.title('relation of iteration times and costs with decreasing learning rate')
```

```
##4.d##
```

```
alpha=0.4
```

```
def logpredict(num,w0,x,y,testx,alpha,lambd):
    cost_batch,w_batch=batchgra(num,x,y,w0,lambd)
    wtx=w_batch*testx.T
    y_pre=[]
    for i in range(0,len(testx)):
        pi=(e**(alpha+float(wtx[... ,i])))/(1+e**(alpha+float(wtx[... ,i])))
        y_pre.append(int(pi+0.5))
    return y_pre
```

```
predict_y=logpredict(3000,w0,x,y,testx,alpha,lambd)
savetxt('wines.csv', predict_y, delimiter = ',')
```

```
#test the accuarcy#
count=0
y_pre1=logpredict(3000,w0,x,y,x,alpha,lambd)
for i in range(0,6000):
    if y_pre1[i]==y[i]:
        count+=1
count
```