Problem5(b) Code Listing

1 (1)(2)

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 19 19:56:04 2018

@author: haofang
"""

from scipy.sparse import csc_matrix
from scipy.sparse.linalg import svds, eigs
import pandas as pd
import numpy as np
import matplotlib . pyplot as plt
#%%
train = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/hw3p5_data/train.txt', sep=",", header=None)
train.columns = ["i", "j", "s"]
test = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/hw3p5_data/test.txt', sep=",", header=None)
test.columns = ["i", "j", "s"]
#%%

def reshape_dataset(index,columns,values,data):
    reshaped_data=data.pivot(index,columns,values).fillna(100)
    return reshaped_data

index="i"
columns="j"
values="s"

reshaped_train=reshape_dataset(index,columns,values,train)
reshaped_test=reshape_dataset(index,columns,values,test)
train=train.drop_duplicates()
test=test.drop_duplicates()

a=train.groupby(['i'],as_index=False)['s'].mean()
b=train.groupby(['j'],as_index=False)['s'].mean()
new_test=test.merge(a, left_on='i', right_on='i', how='inner')
new_test=new_test.merge(b, left_on='j', right_on='j', how='inner')
new_test.columns = ["i", "j", "s","is","js"]
new_test=new_test.sort_values(by=['i'])

```

```
c=test.groupby(['j'],as_index=False).count()
c=c.drop(["s"], axis=1)
c.columns=["j","count"]

new_test=new_test.merge(c,left_on='j', right_on='j', how='inner')


new_test["diff"]=(new_test["js"]*new_test["is"]-new_test["s"])**2


a=new_test['diff'].sum(axis=0)/len(new_test)


new_test["nosquare"]=np.abs(new_test["js"]*new_test["is"]-new_test["s"])
d=new_test.groupby(['j'],as_index=False)["nosquare"].sum()

d=d.merge(c,left_on='j', right_on='j', how='inner')

d.columns=["j","nosquare","count"]
d["divide"]=d["nosquare"]/d["count"]

a=d['divide'].sum(axis=0)/100
mae=a
#reshaped_train=csc_matrix(reshaped_train, dtype=float)


reshaped_test=reshaped_test.values

a=train.groupby(['i'],as_index=False)['s'].mean()
a=a.values
a=a[:,1]
b=train.groupby(['j'],as_index=False)['s'].mean()
b=b.values
b=b[:,1]

test=test.values
collect=0
for u in range(len(a)):
    for v in range(len(b)):
        if reshaped_test[u,v]!=100:
            collect=collect+(np.inner(a[u],b[v])-reshaped_test[u,v])**2
mse=collect/905756

new_reshaped_train=reshaped_train.values
user_ratings_mean = np.mean(new_reshaped_train, axis = 1)
joke_ratings_mean=np.mean(new_reshaped_train,axis=0)
train_demean= new_reshaped_train - user_ratings_mean.reshape(-1, 1)




test = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/hw3p5_data/test.txt', sep=",", header=None)
test.columns = ["i", "j", "s"]
reshaped_test=test.pivot(index,columns,values).fillna(0)
```

```
train = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/hw3p5_data/train.txt', sep=",", header=None)
train.columns = ["i", "j", "s"]
reshaped_train=train.pivot(index,columns,values).fillna(0)
new_reshaped_test=reshaped_test.values
new_reshaped_train=reshaped_train.values




u, s, vt = svds(new_reshaped_train, k=1)
S=np.diag(np.sqrt(s))
U=np.dot(u,S)
V=np.dot(S,vt).T
#ad=s*vt


reshaped_test_100=test.pivot(index,columns,values).fillna(100)

reshaped_train_100=train.pivot(index,columns,values).fillna(100)
new_reshaped_test_100=reshaped_test_100.values
new_reshaped_train_100=reshaped_train_100.values


def MSE(U,V,data_100):
    collect=[]
    for u in range(1000):
        for v in range(500):
            if int(data_100[u,v])!=100:
                collect.append((np.inner(U[u,:],V[v,:])-data_100[u,v])**2)
    return np.mean(collect)
#collect=MSE(U,V,new_reshaped_test_100)



def mae(U,V,data_100):
    collect=[]
    for u in range(1000):
        add=[]
        the_line=data_100[u,:]
        for v in range(500):
            if int(the_line[v])!=100:
                the_one=np.abs((np.inner(U[u,:],V[v,:])-the_line[v]))
                add.append(the_one)
        the_mean=np.mean(add)
        collect.append(the_mean)
    return np.mean(collect)

#mae_error=mae(U,V,new_reshaped_test_100)





#%%
```

```
D=[1,2,5,10,20,50]
test_MSE=[]
train_MSE=[]
test_MAE=[]
train_MAE=[]
for d in D:
    u, s, vt = svds(new_reshaped_train, k=d)
    S=np.diag(np.sqrt(s))
    U=np.dot(u,S)
    V=np.dot(S,vt).T
    testMSE=MSE(U,V,new_reshaped_test_100)
    test_MSE.append(testMSE)
    print("testMSE",testMSE)

    trainMSE=MSE(U,V,new_reshaped_train_100)
    train_MSE.append(trainMSE)
    print("trainMSE",trainMSE)

    trainMAE=mae(U,V,new_reshaped_train_100)
    train_MAE.append(trainMAE)
    print("trainMAE",trainMAE)

    testMAE=mae(U,V,new_reshaped_test_100)
    test_MAE.append(testMAE)
    print("testMAE",testMAE)



#%%
plt.figure()
x=np.arange(1,7)
plt.plot(D,train_MSE,label='Train set')
plt.plot(D,test_MSE,label='Test set')

plt.legend(loc='upper right')
plt.title("MSE of train and test set")
plt.xlabel("Value of 'd'.")
plt.ylabel("MSE")

plt.show()


#%%
plt.figure()
plt.plot(D,train_MAE,label='Train set')
plt.plot(D,test_MAE,label='Test set')

plt.legend(loc='upper right')
plt.title("MAE of train and test set")
plt.xlabel("Value of 'd'.")
plt.ylabel("MAE")

plt.show()
```

2 (3)

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Dec  9 21:01:42 2018

@author: haofang
"""

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Nov 28 13:45:51 2018

@author: haofang
"""
import scipy.sparse
import scipy.sparse . linalg
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.sparse.linalg import svds, eigs

test = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/hw3p5_data/test.txt', sep=",", header=None)
test.columns = ["i", "j", "s"]

train = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/hw3p5_data/train.txt', sep=",", header=None)
train.columns = ["i", "j", "s"]

USER_NUM = train.i.unique().shape[0]
print(USER_NUM)
MOVIE_NUM = train.j.unique().shape[0]
print(MOVIE_NUM)
USER_NUM2 = test.i.unique().shape[0]
print(USER_NUM2)
MOVIE_NUM2 = test.j.unique().shape[0]
print(MOVIE_NUM2)




index='i'
columns='j'
values='s'


def reshape_r_matrix(data,index,columns,values,replace_value):
    reshaped=data.pivot(index,columns,values).fillna(replace_value)
    return reshaped

n=len(train)

n=np.arange(n)
train_label=[]
val_label=[]
```

```
for v in n:
    if v %4==0:
        val_label.append(v)
    else:
        train_label.append(v)
val=train.loc[val_label]
train=train.loc[train_label]




new_train=reshape_r_matrix(train,index,columns,values,0)
new_train.columns=range(new_train.shape[1])
new_train.index=range(new_train.shape[0])

new_val=reshape_r_matrix(val,index,columns,values,0)
new_val.columns=range(new_val.shape[1])
new_val.index=range(new_val.shape[0])


new_test=reshape_r_matrix(test,index,columns,values,0)
new_test.columns=range(new_test.shape[1])
new_test.index=range(new_test.shape[0])


def MSE(U,V,data_100):
    collect=[]
    for u in range(USER_NUM):
         for v in range(MOVIE_NUM ):
               if int(data_100[u,v])!=100:
                     collect.append((np.inner(U[u,:],V[:,v])-data_100[u,v])**2)
    return np.mean(collect)

def mae(U,V,data_100):
    collect=[]
    for u in range(USER_NUM):
        add=[]
        the_line=data_100[u,:]
        for v in range(MOVIE_NUM ):
            if int(the_line[v])!=100:
                the_one=np.abs((np.inner(U[u,:],V[:,v])-the_line[v]))
                add.append(the_one)
        the_mean=np.mean(add)
        collect.append(the_mean)
    return np.mean(collect)


#
def iterate_UV(lam,k,data):
     n=data.shape[0]
     m=data.shape[1]
     old_u = np.random.randn(n, k)
     old_v = np.random.randn(k, m)
     lam=lam*np.eye(k)
     for v in range(30):
            new_u = np.zeros((n, k))
            new_v = np.zeros((k, m))
            for i in range(n):
                 a=data.loc[i].nonzero()[0]
                 b=data.loc[i]
                 b=b[a]
                 picked_old_v=old_v[:,a]
                 the_u = np.linalg.solve(np.dot(picked_old_v, picked_old_v.T)+lam, np.dot(picked_old_v,b))
                 new_u[i,:] = the_u.T
            for i in range(m):
                 c=data[i].nonzero()[0]
                 d=data[i]
                 d=d[c]
                 picked_old_u=new_u[c,:]
                 a=np.dot(picked_old_u.T,picked_old_u)+lam
                 b=np.dot(picked_old_u.T,d)
                 the_v= np.linalg.solve(a,b)
                 new_v[:,i] = the_v
            old_u=new_u
            old_v=new_v

     return old_u,old_v
#u,v=iterate_UV(0.2,10,new_train)

#%%


#%%
good_lamda=[]
All_D=[1,2,5,10,20,50]
All_lamda=[]
for a in range(-8,6):
    All_lamda.append(2**a)
error=[]



#%%
reshaped_val=val.pivot(index,columns,values).fillna(100)
new_reshaped_val=reshaped_val.values

train_table=[]
for d in All_D:
    result=[]
    for lam in All_lamda:
        u,v=iterate_UV(lam,d,new_train)
        print(d)
        print(lam)
        the_val_error=mae(u,v,new_reshaped_val)
        result.append(the_val_error)
    train_table.append(result)

#%%
good_lambdas=[]
for v in range(len(train_table)):
    index=np.argmin(train_table[v])
    good_lambdas.append(All_lamda[index])

#%%

test = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/hw3p5_data/test.txt', sep=",", header=None)
test.columns = ["i", "j", "s"]

train = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/hw3p5_data/train.txt', sep=",", header=None)
train.columns = ["i", "j", "s"]



index='i'
columns='j'
values='s'

reshaped_train=train.pivot(index,columns,values).fillna(100)
reshaped_test=test.pivot(index,columns,values).fillna(100)
new_reshaped_test=reshaped_test.values
new_reshaped_train=reshaped_train.values


new_train=reshape_r_matrix(train,index,columns,values,0)
new_train.columns=range(new_train.shape[1])
new_train.index=range(new_train.shape[0])

new_test=reshape_r_matrix(test,index,columns,values,0)
new_test.columns=range(new_test.shape[1])
new_test.index=range(new_test.shape[0])



    #%%
good_lambdas=[0.0625,2,4,4,8,8]


test_MSE=[]
train_MSE=[]
test_MAE=[]
train_MAE=[]
for v in range(len(All_D)):
    d=All_D[v]
    lam=good_lambdas[v]
    d=All_D[v]
    the_train_error=0
    the_test_error=0
    u,v=iterate_UV(lam,d,new_train)
    print(d)
    testMSE=MSE(u,v,new_reshaped_test)
    test_MSE.append(testMSE)
    print("testMSE",testMSE)

    trainMSE=MSE(u,v,new_reshaped_train)
    train_MSE.append(trainMSE)
    print("trainMSE",trainMSE)

    trainMAE=mae(u,v,new_reshaped_train)
    train_MAE.append(trainMAE)
    print(”trainMAE”,trainMAE)

    testMAE=mae(u,v,new_reshaped_test)
    test_MAE.append(testMAE)
    print(”testMAE”,testMAE)


#%%
D=[1,2,5,10,20,50]
plt.figure()
x=np.arange(1,7)
plt.plot(D,train_MSE,label='Train set')
plt.plot(D,test_MSE,label='Test set')

plt.legend(loc='lower left')
plt.title(”MSE of train and test set”)
plt.xlabel(”Value of 'd'.”)
plt.ylabel(”MSE”)

plt.show()



plt.figure()
plt.plot(D,train_MAE,label='Train set')
plt.plot(D,test_MAE,label='Test set')

plt.legend(loc='lower left')
plt.title(”MAE of train and test set”)
plt.xlabel(”Value of 'd'.”)
plt.ylabel(”MAE”)

plt.show()
```