

## Problem5(b) Code Listing

### 1 (1)(2)

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Mon Nov 19 19:56:04 2018
5
6 @author: haofang
7 """
8
9 from scipy.sparse import csc_matrix
10 from scipy.sparse.linalg import svds, eigs
11 import pandas as pd
12 import numpy as np
13 import matplotlib.pyplot as plt
14 #%%
15 train = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/
16 hw3p5_data/train.txt', sep=",", header=None)
17 train.columns = ["i", "j", "s"]
18 test = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/
19 hw3p5_data/test.txt', sep=",", header=None)
20 test.columns = ["i", "j", "s"]
21 #%%
22 def reshape_dataset(index, columns, values, data):
23     reshaped_data = data.pivot(index, columns, values).fillna(100)
24     return reshaped_data
25
26 index="i"
27 columns="j"
28 values="s"
29
30 reshaped_train = reshape_dataset(index, columns, values, train)
31 reshaped_test = reshape_dataset(index, columns, values, test)
32 train = train.drop_duplicates()
33 test = test.drop_duplicates()
34
35 a = train.groupby(['i'], as_index=False)['s'].mean()
36 b = train.groupby(['j'], as_index=False)['s'].mean()
37 new_test = test.merge(a, left_on='i', right_on='i', how='inner')
38 new_test = new_test.merge(b, left_on='j', right_on='j', how='inner')
39 new_test.columns = ["i", "j", "s", "is", "js"]
40 new_test = new_test.sort_values(by=['i'])
```

```

41 c=test.groupby(['j'],as_index=False).count()
42 c=c.drop(["s"], axis=1)
43 c.columns=["j","count"]
44
45 new_test=new_test.merge(c,left_on='j', right_on='j', how='inner')
46
47
48 new_test["diff"]=(new_test["js"]*new_test["is"]-new_test["s"])**2
49
50
51 a=new_test['diff'].sum(axis=0)/len(new_test)
52
53
54 new_test["nosquare"]=np.abs(new_test["js"]*new_test["is"]-new_test[
    "s"])
55 d=new_test.groupby(['j'],as_index=False)["nosquare"].sum()
56
57 d=d.merge(c,left_on='j', right_on='j', how='inner')
58
59 d.columns=["j","nosquare","count"]
60 d["divide"]=d["nosquare"]/d["count"]
61
62 a=d['divide'].sum(axis=0)/100
63 mae=a
64 #reshaped_train=csc_matrix(reshaped_train, dtype=float)
65
66
67 reshaped_test=reshaped_test.values
68
69 a=train.groupby(['i'],as_index=False)['s'].mean()
70 a=a.values
71 a=a[:,1]
72 b=train.groupby(['j'],as_index=False)['s'].mean()
73 b=b.values
74 b=b[:,1]
75
76 test=test.values
77 collect=0
78 for u in range(len(a)):
79     for v in range(len(b)):
80         if reshaped_test[u,v]!=100:
81             collect=collect+(np.inner(a[u],b[v])-reshaped_test[u,v]
82             ])**2
83 mse=collect/905756
84
85 new_reshaped_train=reshaped_train.values
86 user_ratings_mean = np.mean(new_reshaped_train, axis = 1)
87 joke_ratings_mean=np.mean(new_reshaped_train,axis=0)
88 train_demean= new_reshaped_train - user_ratings_mean.reshape(-1, 1)
89
90
91
92 test = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/
    hw3p5_data/test.txt', sep=",", header=None)
93 test.columns = ["i", "j", "s"]
94 reshaped_test=test.pivot(index,columns,values).fillna(0)

```

```

95 train = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/
    hw3p5_data/train.txt', sep=",", header=None)
96 train.columns = ["i", "j", "s"]
97 reshaped_train=train.pivot(index,columns,values).fillna(0)
98 new_resaped_test=reshaped_test.values
99 new_resaped_train=reshaped_train.values
100
101
102
103
104 u, s, vt = svds(new_resaped_train, k=1)
105 S=np.diag(np.sqrt(s))
106 U=np.dot(u,S)
107 V=np.dot(S,vt).T
108 #ad=s*vt
109
110
111 reshaped_test_100=test.pivot(index,columns,values).fillna(100)
112
113 reshaped_train_100=train.pivot(index,columns,values).fillna(100)
114 new_resaped_test_100=reshaped_test_100.values
115 new_resaped_train_100=reshaped_train_100.values
116
117
118 def MSE(U,V,data_100):
119     collect=[]
120     for u in range(1000):
121         for v in range(500):
122             if int(data_100[u,v])!=100:
123                 collect.append((np.inner(U[u,:],V[v,:])-
124                     data_100[u,v])**2)
125     return np.mean(collect)
126 #collect=MSE(U,V,new_resaped_test_100)
127
128
129 def mae(U,V,data_100):
130     collect=[]
131     for u in range(1000):
132         add=[]
133         the_line=data_100[u,:]
134         for v in range(500):
135             if int(the_line[v])!=100:
136                 the_one=np.abs((np.inner(U[u,:],V[v,:])-the_line[v]
137                     ))
138                 add.append(the_one)
139             the_mean=np.mean(add)
140             collect.append(the_mean)
141     return np.mean(collect)
142 #mae_error=mae(U,V,new_resaped_test_100)
143
144
145
146
147
148 #%%

```

```

149 D=[1,2,5,10,20,50]
150 test_MSE=[]
151 train_MSE=[]
152 test_MAE=[]
153 train_MAE=[]
154 for d in D:
155     u, s, vt = svds(new_resaped_train, k=d)
156     S=np.diag(np.sqrt(s))
157     U=np.dot(u,S)
158     V=np.dot(S,vt).T
159     testMSE=MSE(U,V,new_resaped_test_100)
160     test_MSE.append(testMSE)
161     print("testMSE",testMSE)
162
163     trainMSE=MSE(U,V,new_resaped_train_100)
164     train_MSE.append(trainMSE)
165     print("trainMSE",trainMSE)
166
167     trainMAE=mae(U,V,new_resaped_train_100)
168     train_MAE.append(trainMAE)
169     print("trainMAE",trainMAE)
170
171     testMAE=mae(U,V,new_resaped_test_100)
172     test_MAE.append(testMAE)
173     print("testMAE",testMAE)
174
175
176
177 ###
178 plt.figure()
179 x=np.arange(1,7)
180 plt.plot(D,train_MSE,label='Train set')
181 plt.plot(D,test_MSE,label='Test set')
182
183 plt.legend(loc='upper right')
184 plt.title("MSE of train and test set")
185 plt.xlabel("Value of 'd'.")
186 plt.ylabel("MSE")
187
188 plt.show()
189
190
191 ###
192 plt.figure()
193 plt.plot(D,train_MAE,label='Train set')
194 plt.plot(D,test_MAE,label='Test set')
195
196 plt.legend(loc='upper right')
197 plt.title("MAE of train and test set")
198 plt.xlabel("Value of 'd'.")
199 plt.ylabel("MAE")
200
201 plt.show()

```

2 (3)

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Sun Dec  9 21:01:42 2018
5
6  @author: haofang
7  """
8
9  #!/usr/bin/env python3
10 # -*- coding: utf-8 -*-
11 """
12 Created on Wed Nov 28 13:45:51 2018
13
14 @author: haofang
15 """
16 import scipy.sparse
17 import scipy.sparse . linalg
18 import numpy as np
19 import pandas as pd
20 import matplotlib.pyplot as plt
21 from scipy.sparse.linalg import svds, eigs
22
23 test = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/
24                  hw3p5_data/test.txt', sep=",", header=None)
25 test.columns = ["i", "j", "s"]
26
27 train = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/
28                    hw3p5_data/train.txt', sep=",", header=None)
29 train.columns = ["i", "j", "s"]
30
31 USER_NUM = train.i.unique().shape[0]
32 print(USER_NUM)
33 MOVIE_NUM = train.j.unique().shape[0]
34 print(MOVIE_NUM)
35 USER_NUM2 = test.i.unique().shape[0]
36 print(USER_NUM2)
37 MOVIE_NUM2 = test.j.unique().shape[0]
38 print(MOVIE_NUM2)
39
40
41 index='i'
42 columns='j'
43 values='s'
44
45
46 def reshape_r_matrix(data,index,columns,values,replace_value):
47     reshaped=data.pivot(index,columns,values).fillna(replace_value)
48     return reshaped
49
50 n=len(train)
51
52 n=np.arange(n)
53 train_label=[]
54 val_label=[]

```

```

55 for v in n:
56     if v %4==0:
57         val_label.append(v)
58     else:
59         train_label.append(v)
60 val=train.loc[val_label]
61 train=train.loc[train_label]
62
63
64
65
66 new_train=reshape_r_matrix(train,index,columns,values,0)
67 new_train.columns=range(new_train.shape[1])
68 new_train.index=range(new_train.shape[0])
69
70 new_val=reshape_r_matrix(val,index,columns,values,0)
71 new_val.columns=range(new_val.shape[1])
72 new_val.index=range(new_val.shape[0])
73
74
75 new_test=reshape_r_matrix(test,index,columns,values,0)
76 new_test.columns=range(new_test.shape[1])
77 new_test.index=range(new_test.shape[0])
78
79
80 def MSE(U,V,data_100):
81     collect=[]
82     for u in range(USER_NUM):
83         for v in range(MOVIE_NUM):
84             if int(data_100[u,v])!=100:
85                 collect.append((np.inner(U[u,:],V[:,v])-
86 data_100[u,v])**2)
87     return np.mean(collect)
88
89 def mae(U,V,data_100):
90     collect=[]
91     for u in range(USER_NUM):
92         add=[]
93         the_line=data_100[u,:]
94         for v in range(MOVIE_NUM):
95             if int(the_line[v])!=100:
96                 the_one=np.abs((np.inner(U[u,:],V[:,v])-the_line[v]
97 ))
98                 add.append(the_one)
99                 the_mean=np.mean(add)
100                 collect.append(the_mean)
101     return np.mean(collect)
102
103 #
104 def iterate_UV(lam,k,data):
105     n=data.shape[0]
106     m=data.shape[1]
107     old_u = np.random.randn(n, k)
108     old_v = np.random.randn(k, m)
109     lam=lam*np.eye(k)
110     for v in range(30):

```

```

110         new_u = np.zeros((n, k))
111         new_v = np.zeros((k, m))
112         for i in range(n):
113             a=data.loc[i].nonzero()[0]
114             b=data.loc[i]
115             b=b[a]
116             picked_old_v=old_v[:,a]
117             the_u = np.linalg.solve(np.dot(picked_old_v ,
picked_old_v.T)+lam, np.dot(picked_old_v ,b))
118             new_u[i,:] = the_u.T
119         for i in range(m):
120             c=data[i].nonzero()[0]
121             d=data[i]
122             d=d[c]
123             picked_old_u=new_u[c,:]
124             a=np.dot(picked_old_u.T,picked_old_u)+lam
125             b=np.dot(picked_old_u.T,d)
126             the_v= np.linalg.solve(a,b)
127             new_v[:,i] = the_v
128         old_u=new_u
129         old_v=new_v
130
131         return old_u , old_v
132 #u,v=iterate_UV(0.2,10,new_train)
133
134 ###
135
136
137 ###
138 good_lamda=[]
139 All_D=[1,2,5,10,20,50]
140 All_lamda=[]
141 for a in range(-8,6):
142     All_lamda.append(2**a)
143 error=[]
144
145
146
147 ###
148 reshaped_val=val.pivot(index , columns , values).fillna(100)
149 new_reshaped_val=reshaped_val.values
150
151 train_table=[]
152 for d in All_D:
153     result=[]
154     for lam in All_lamda:
155         u,v=iterate_UV(lam,d,new_train)
156         print(d)
157         print(lam)
158         the_val_error=mae(u,v,new_reshaped_val)
159         result.append(the_val_error)
160     train_table.append(result)
161
162 ###
163 good_lambdas=[]
164 for v in range(len(train_table)):
165     index=np.argmin(train_table[v])

```

```

166     good_lambdas.append(All_lambda[index])
167
168     ###
169
170     test = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/
171         hw3p5_data/test.txt', sep=",", header=None)
172     test.columns = ["i", "j", "s"]
173
174     train = pd.read_csv('/Users/haofang/Desktop/UW/cse546/hw3re/
175         hw3p5_data/train.txt', sep=",", header=None)
176     train.columns = ["i", "j", "s"]
177
178     index='i'
179     columns='j'
180     values='s'
181
182     reshaped_train=train.pivot(index,columns,values).fillna(100)
183     reshaped_test=test.pivot(index,columns,values).fillna(100)
184     new_resaped_test=reshaped_test.values
185     new_resaped_train=reshaped_train.values
186
187
188     new_train=reshape_r_matrix(train,index,columns,values,0)
189     new_train.columns=range(new_train.shape[1])
190     new_train.index=range(new_train.shape[0])
191
192     new_test=reshape_r_matrix(test,index,columns,values,0)
193     new_test.columns=range(new_test.shape[1])
194     new_test.index=range(new_test.shape[0])
195
196
197
198     ###
199     good_lambdas=[0.0625,2,4,4,8,8]
200
201
202     test_MSE=[]
203     train_MSE=[]
204     test_MAE=[]
205     train_MAE=[]
206     for v in range(len(All_D)):
207         d=All_D[v]
208         lam=good_lambdas[v]
209         d=All_D[v]
210         the_train_error=0
211         the_test_error=0
212         u,v=iterate_UV(lam,d,new_train)
213         print(d)
214         testMSE=MSE(u,v,new_resaped_test)
215         test_MSE.append(testMSE)
216         print("testMSE",testMSE)
217
218         trainMSE=MSE(u,v,new_resaped_train)
219         train_MSE.append(trainMSE)
220         print("trainMSE",trainMSE)

```



```

221
222     trainMAE=mae(u,v,new_resaped_train)
223     train_MAE.append(trainMAE)
224     print("trainMAE",trainMAE)
225
226     testMAE=mae(u,v,new_resaped_test)
227     test_MAE.append(testMAE)
228     print("testMAE",testMAE)
229
230
231     ###
232     D=[1,2,5,10,20,50]
233     plt.figure()
234     x=np.arange(1,7)
235     plt.plot(D,train_MSE,label='Train set')
236     plt.plot(D,test_MSE,label='Test set')
237
238     plt.legend(loc='lower left')
239     plt.title("MSE of train and test set")
240     plt.xlabel("Value of 'd'.")
241     plt.ylabel("MSE")
242
243     plt.show()
244
245
246
247     plt.figure()
248     plt.plot(D,train_MAE,label='Train set')
249     plt.plot(D,test_MAE,label='Test set')
250
251     plt.legend(loc='lower left')
252     plt.title("MAE of train and test set")
253     plt.xlabel("Value of 'd'.")
254     plt.ylabel("MAE")
255
256     plt.show()

```