

Homework #4 - B

Spring 2020, CSE 446/546: Machine Learning

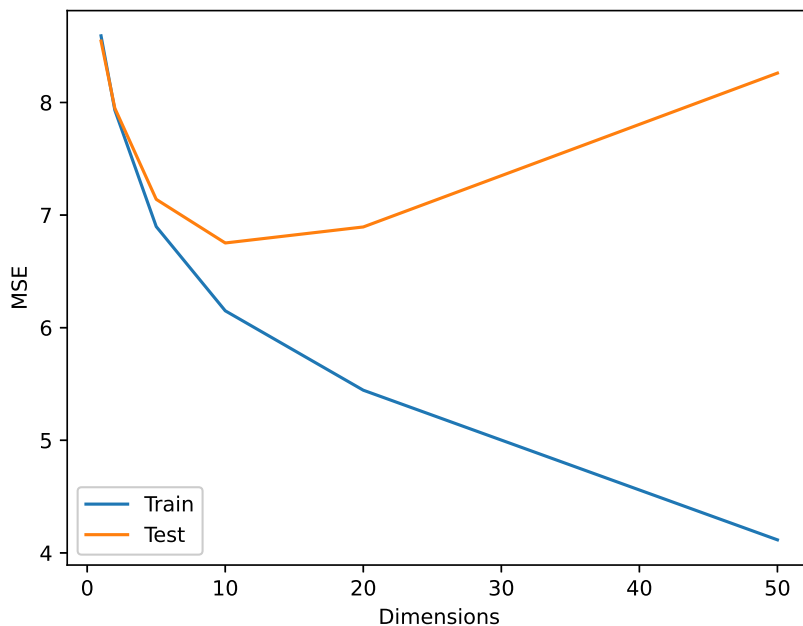
Richy Yun

Due: Thursday 5/28/2020 11:59 PM

Matrix Completion and Recommendation System

B.1

- a. [5 points] For this estimator we simply assume all users' ratings are the average rating of each movie. Thus μ is the estimation for each movie for every user. With this estimator we get $\mathcal{E}_{test}(\hat{R}) = 3.0882$
- b. [5 points] The errors are shown below.



- c. [10 points]
- d. [10 points]
- e. [5 points]
- f. [5 points]

Source Code A.3

```
import csv
import numpy as np
from scipy.sparse.linalg import svds
import matplotlib.pyplot as plt

data = []

with open('ml-100k\\u.data') as csvfile:
    spamreader = csv.reader(csvfile, delimiter='\t')
    for row in spamreader:
        data.append([int(row[0])-1, int(row[1])-1, int(row[2])])
data = np.array(data)

num_observations = len(data)    # num_observations = 100,000
num_users = max(data[:, 0])+1  # num_users = 943, indexed 0,...,942
num_items = max(data[:, 1])+1  # num_items = 1682 indexed 0,...,1681

np.random.seed(1)
num_train = int(0.8*num_observations)
perm = np.random.permutation(data.shape[0])
train = data[perm[0:num_train], :]
test = data[perm[num_train:], :]

def mse(a, b):
    return np.mean(np.square(a-b))

"""
a)
"""
# Calculate average
avgScore = np.zeros(num_items)
trainItems = train[:, 1]
trainItems = trainItems[np.unique(trainItems)]
for i in range(len(trainItems)):
    avgScore[trainItems[i]] = np.mean(train[train[:, 1] == trainItems[i], 2])

# Do prediction
predictions = avgScore[test[:, 1]]
error = mse(predictions, test[:, 2])

"""
b)
"""
# Set up R tilde
Rtrain = np.zeros((num_items, num_users))
for i in range(len(train)):
    Rtrain[train[i, 1], train[i, 0]] = train[i, 2]

# Do SVD and predictions
d = [1, 2, 5, 10, 20, 50]
trainerr = np.zeros(len(d))
testerr = np.zeros(len(d))
for i in range(len(d)):
    u, s, vt = svds(Rtrain, k=d[i])
```

```

reconst = np.matmul(u*s, vt)
trainpred = reconst[train[:, 1], train[:, 0]]
trainerr[i] = mse(train[:, 2], trainpred)
testpred = reconst[test[:, 1], test[:, 0]]
testerr[i] = mse(test[:, 2], testpred)

# Plot
plt.plot(d, trainerr)
plt.plot(d, testerr)
plt.xlabel('Dimensions')
plt.ylabel('MSE')
plt.legend(('Train', 'Test'))

```