task2

February 28, 2017

In [ ]:
```
# last submit: 4.5, 9.3
```

In [2]:
```
import gzip
from collections import defaultdict
import math
import scipy.optimize
from sklearn import svm
import numpy
import string

def readGz(f):
    for l in gzip.open(f):
        yield eval(l)
```

In [3]:
```
data = []
for l in readGz("assignment1/train.json.gz"):
    data.append(l)

data_train = data[:100000]
data_valid = data[100000:]
UserRating = defaultdict(list)
ItemRating = defaultdict(list)
for r in data:
    UserRating[r['reviewerID']].append(r)
    ItemRating[r['itemID']].append(r)
```

In [4]:
```
trainRatings = [r['rating'] for r in data]
globalAverage = sum(trainRatings) * 1.0 / len(trainRatings)

betaU = {}
betaI = {}
for u in UserRating:
    betaU[u] = 0

for i in ItemRating:
    betaI[i] = 0

alpha = globalAverage
```

In [5]:
```
def iterate(lamU, lamI):
    # update alpha
    newAlpha = 0
    for r in data:
        newAlpha += r['rating'] - (betaU[r['reviewerID']] + betaI[r['itemID']])
    alpha = newAlpha / len(data)

    # update betaU
    for u in UserRating:
        newBetaU = 0
        for r in UserRating[u]:
            newBetaU += r['rating'] - (alpha + betaI[r['itemID']])
        betaU[u] = newBetaU / (lamU + len(UserRating[u]))

    # update betaI
    for i in ItemRating:
        newBetaI = 0
        for r in ItemRating[i]:
            newBetaI += r['rating'] - (alpha + betaU[r['reviewerID']])
        betaI[i] = newBetaI / (lamI + len(ItemRating[i]))

    # cal mse
    mse = 0
    for r in data:
        predict = alpha + betaU[r['reviewerID']] + betaI[r['itemID']]
        mse += (r['rating'] - predict)**2

    # add regularizer
    regU = 0
    regI = 0
    for u in betaU:
        regU += betaU[u]**2
    for i in betaI:
        regI += betaI[i]**2

    mse /= len(data)
    return mse, mse + lamU*regU + lamI*regI
```

In [6]:
```
# lamU = 4.5
# lamI = 9.3
# MSE = 0.806253035
# iteration = 30

mse,objective = iterate(1,1)
newMSE,newObjective = iterate(1,1)

n = 1
while n < 30 or objective - newObjective > 0.0001:
```

```
            mse, objective = newMSE, newObjective
            newMSE, newObjective = iterate(4.5, 9.3)
            n += 1

        validMSE = 0
        for r in data_valid:
            bu = 0
            bi = 0
            if r['reviewerID'] in betaU:
                bu = betaU[r['reviewerID']]
            if r['itemID'] in betaI:
                bi = betaI[r['itemID']]
            prediction = alpha + bu + bi
            validMSE += (r['rating'] - prediction)**2

        validMSE /= len(data_valid)
        print("MSE = " + str(validMSE))
```

```
MSE = 0.806253035
```

In [9]:
```
predictions = open("assignment1/predictions_Rating.txt", 'w')
for l in open("assignment1/pairs_Rating.txt"):
    if l.startswith("userID"):
        #header
        predictions.write(l)
        continue
    u,i = l.strip().split('-')

    x = alpha
    if u in betaU:
        x += betaU[u]
    if i in betaI:
        x += betaI[i]
    predictions.write(u + '-' + i + ',' + str(x) + '\n')

predictions.close()
```