IS643 Final Project Planning Document

Alex Satz

April 16 2017

# Summary Statement: Optimize processing time of my previously submitted recommender system.

Background: The functions below are used to ‘normalize’ my recommender system, for both the item- and user-based CF, leading to excellent results (RMSE < 1). However, the normalization process is very slow, requiring 5-10 minutes of computational time.

Goal for the final project: I would like to optimize the recommender, and find an alternative method with faster speeds with minimal increase in RMSE. With this optimization, the recommender system that has been in development since the beginning of the class may be considered completed.

def normalize2(pred, normMatrix):

'''by norm sim value to 1; only using cases where dot product is non-zero'''

for row in range(0, pred.shape[0]):

for col in range(0, pred.shape[1]):

if pred[row,col] != 0:

pred[row,col] = pred[row,col]\*normMatrix[row,col]

return pred

def getNormMatrix(similarity, ratings):

l1 = np.zeros((ratings.shape[0], ratings.shape[1]))

for c in range (0, similarity.shape[0]):

for r in range (0, ratings.shape[0]):

new = similarity[c]\*ratings[r]

i = np.nonzero(new)

nzero = np.sum(similarity[c][i])

if nzero == 0:

normfactor = 1

else:

normfactor = 1.0/nzero

l1[r,c] = normfactor

return l1

def predict4(ratings, similarity, type='user'):

'''normalized by sum of similarity values associated with non-zero ratings and normalize to 1'''

if type == 'user':

pass

elif type == 'item':

pred = ratings.dot(similarity)

normMatrix = getNormMatrix(similarity, ratings)

pred = normalize2(pred, normMatrix)

return pred