Q2 $\nabla Q(w) = \sum_{i=1}^{n} \nabla_w Q_i(w)$.

Here $i \rightarrow$ denotes the sample number

$\nabla_w Q_i(w) \rightarrow$ It is the gradient with respect to sample 'i'

and $\Delta Q(w) \rightarrow$ It is the full gradient after summation of gradient of all the samples.

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = \left(r_{ij} - \sum_{k=1}^{k} p_{ik} q_{kj}\right)^2 + \frac{\gamma}{2}\left(\|P\|^2 + \|Q\|^2\right)$$

$$\frac{d}{dp_{ik}} e_{ij}^2 = -2 (r_{ij} - \hat{r}_{ij})(q_{kj}) = -2 e_{ij} q_{kj} + \gamma p_{ik}$$

$$\frac{d}{dq_{kj}} e_{ij}^2 = -2 (r_{ij} - \hat{r}_{ij}) p_{ik} = -2 e_{ij} p_{ik} + \gamma q_{kj}$$

Full gradient vector

$$\nabla L(P,Q) = \begin{bmatrix} \nabla_{p_1}(L(p_1,q_1)) \\ \nabla_{p_2}(L(p_2,q_1)) \\ \vdots \\ \nabla_{p_m}(L(p_m,q_k)) \\ \vdots \\ \nabla_{q_1}(L(p_1,q_1)) \\ \nabla_{q_2}(L(p_1,q_2)) \\ \vdots \\ \nabla_{q_n}(L(p_1,q_n)) \end{bmatrix}$$

Q1 The error function is given by

$$e_{ij}^2 = (r_{ij} - \text{prediction at } (i,j))^2$$

$$= \left(r_{ij} - \sum_{k=1}^{k=k} p_{ik} q_{kj}\right)^2$$

This needs to be modified to prevent overfitting.

So, penalization for large model $\|P\|$ and $\|Q\|$ need to be done. Hence, a regularisation term is added :-

$$\gamma/2 \left(\|P\|_F^2 + \|Q\|_F^2\right)$$

So, new error term is given by

$$\tilde{e}_{ij} = e_{ij}^2 + \gamma/2 \left(\|P\|^2 + \|Q\|^2\right)$$

$$\Rightarrow \tilde{e}_{ij} = \left(r_{ij} - \sum_{s=1}^{k} p_{is} q_{sj}\right)^2 + \gamma/2 \left(\|P\|^2 + \|Q\|^2\right)$$

$$\Rightarrow L(\cdot) = \sum_{(i,j) \in K} \tilde{e}_{ij}$$

The Minimization of this function

would ensure that the term

$$\left(r_{ij} - \sum_{s=1}^{s=k} p_{is} q_{sj}\right)^2 \text{ is minimised.}$$

Minimisation of this term would mean that $\sum_{s=1}^{k} p_{is} q_{sj}$ would give a nice approximation for the value r_{ij}. Also, the terms $\|P\|^2$ & $\|Q\|^2$ would penalise overfitting.

⇒ High 'γ' increases the penalty and prevents overfitting.

Q2 $e_{ij} = (r_{ij} - \sum p_{i*} q_{*j})$

$$\tilde{e}_{ij} = \left(r_{ij} - \sum_{s=1}^{k} p_{is} q_{sj}\right)^2 + \gamma/2\left(\|P\|^2 + \|Q\|^2\right)$$

and $L(\cdot) = \sum_{(i,j)\in K} \tilde{e}_{ij}$

So, for a sample (i,j)

$$\frac{d\,\tilde{e}_{ij}}{dp_{is}} = \frac{d\left(r_{ij} - \sum_{s=1}^{k} p_{is} q_{sj}\right)^2 + \gamma/2\left(\|P\|^2 + \|Q\|^2\right)}{dp_{is}}$$

$$= -2\left(r_{ij} - \sum_{s=1}^{k} p_{is} q_{sj}\right)\left(\frac{d\sum_{s=1}^{k} p_{is} q_{sj}}{dp_{is}}\right) + 2\frac{\gamma}{2}(p_{is})$$

$$\frac{d\tilde{e}_{ij}}{dp_{is}} = -2 e_{ij} q_{sj} + \gamma p_{is}$$

$$\frac{d\tilde{e}_{ij}}{dq_{sj}} = -2 e_{ij} p_{is} + \gamma q_{sj}$$

So, all the p_{is}'s where $s = 1 \ldots k$ and q_{sj} where $s = 1 \ldots k$ would be updated only by differentiating a single term e_{ij}^2.

So, the update :-

$$(p_{i+1*}, q_{*j+1})^T = (p_{i*}, q_{*j})^T + \alpha \left(2e_{ij} q_{*j} - \gamma p_{i*}\right), \left(2e_{ij} p_{i*} - \gamma q_{*j}\right)^T$$

is stochastic

Question-2b: Read the main reference paper and justify why bias is added. Note in this notebook we add global bias. Which other bias term was suggested in the paper?

Answer:

1. The purpose of adding bias is to keep a balance between overfitting and underfitting. The model's variance can be changed by adding bias. The purpose for adding a bias value is to fit the data better. There can be a possibility that model would hyper personalise the rating for the user and lead to overfiiting. So, it would only provide correct results for people in the training set and would lead to inaccurate results in new data. variance van't be decreased without increasing the bias. So, in order to make the model more general such that it works for all kind of data, bias needs to be added.
2. The other bias terms that were suggested in the paper were:

   The observed variation in rating values is due to effects associated with either users or items, known as biases or intercepts, independent of any interactions. For example, typical collaborative filtering data exhibits large systematic tendencies for some users to give higher ratings than others, and for some items to receive higher ratings than others. After all, some products are widely perceived as better (or worse) than others. The other bias terms are b_i and b_u which are item and user bias respectively apart from the bias already defined which is , u, i.e. global average.

   So, $r_{ui} = b_i + b_u + q_i^T p_u + u$

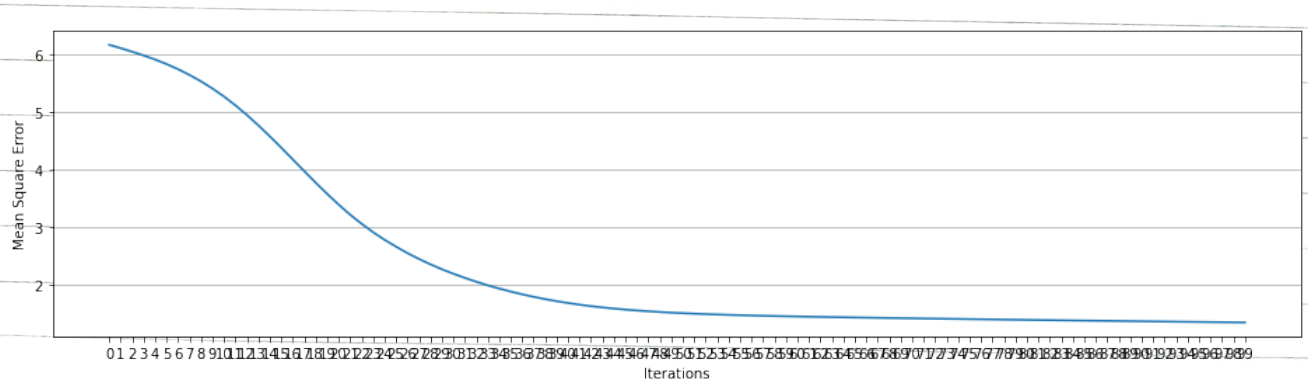
   b_i denotes the item bias which is the value by which average rating of an item differs from other items

   b_u denotes the user bias which is the value by which average rating of a user differs from other users

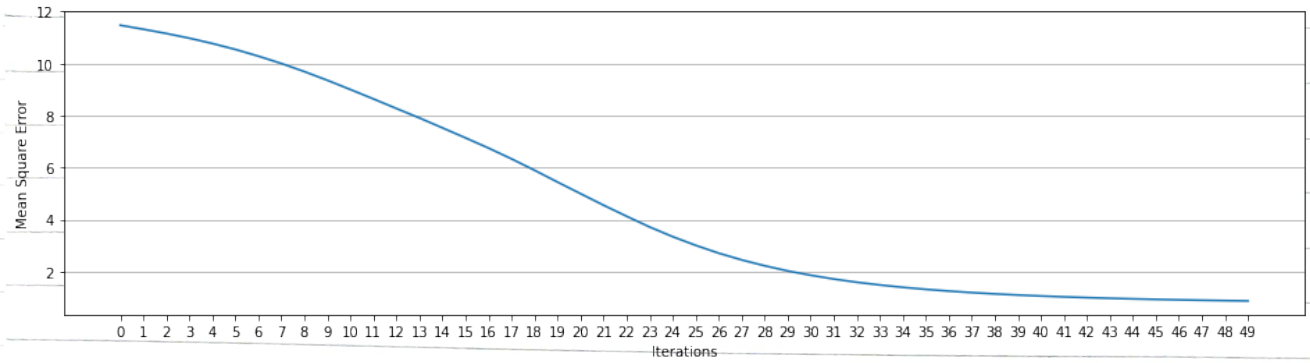
   u is the average rating from the entire matrix

   r_{ui} is the predicted rating.

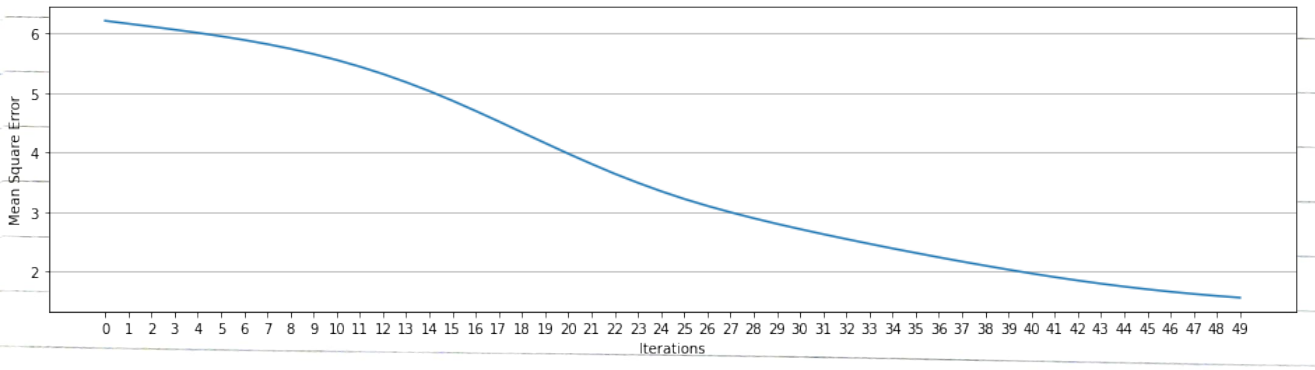
Q5 Graph



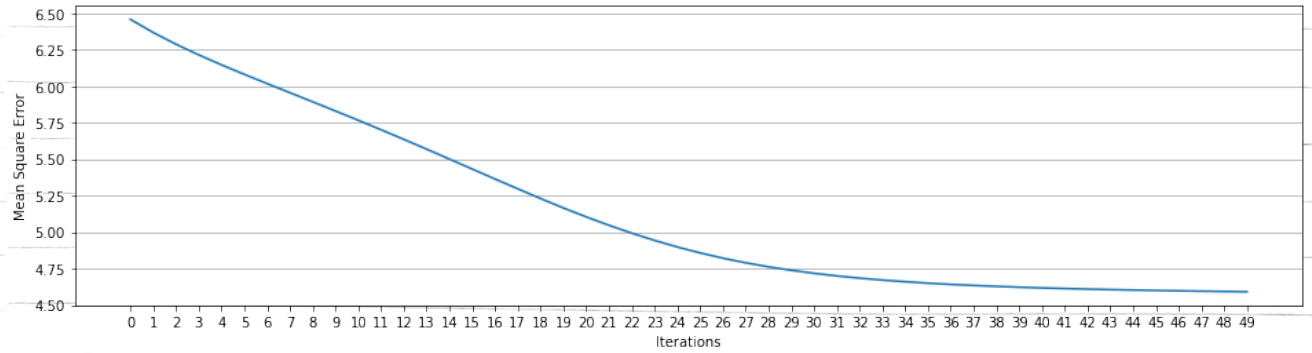
Q6 graph



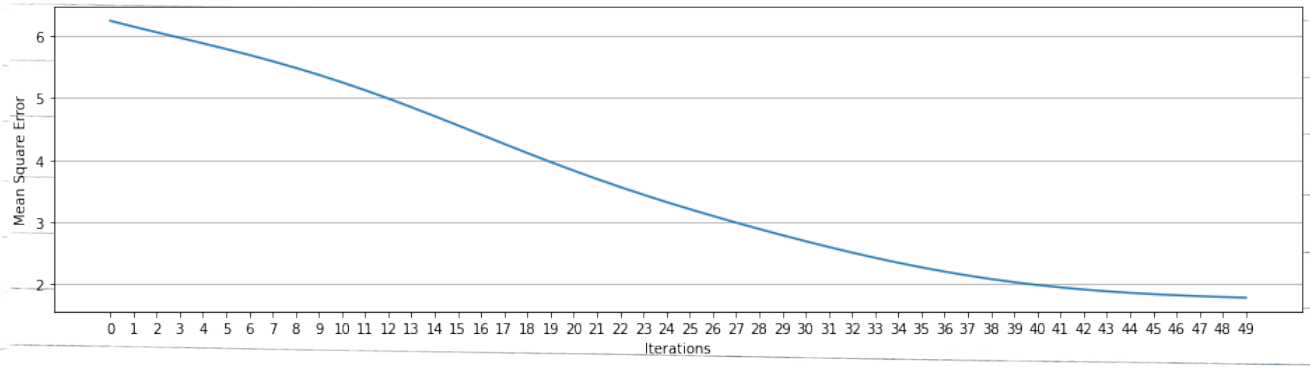
Q7 graph



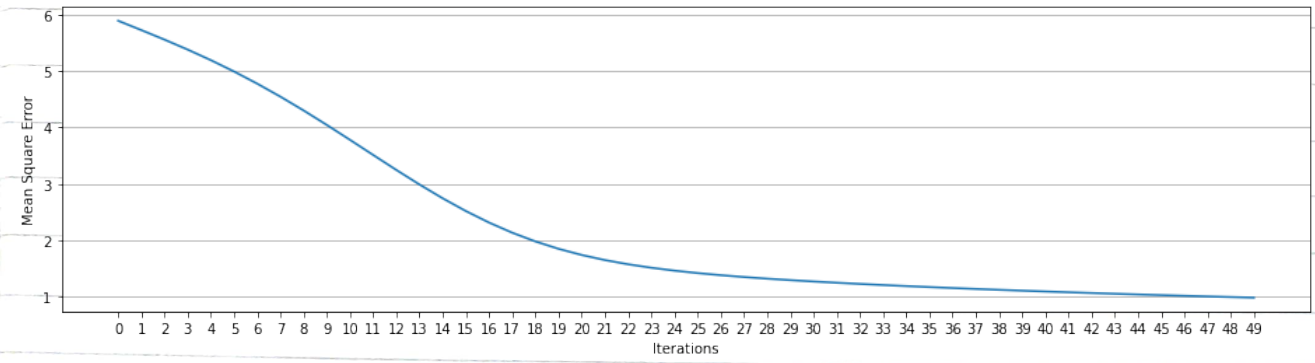
Q8 gamma=1



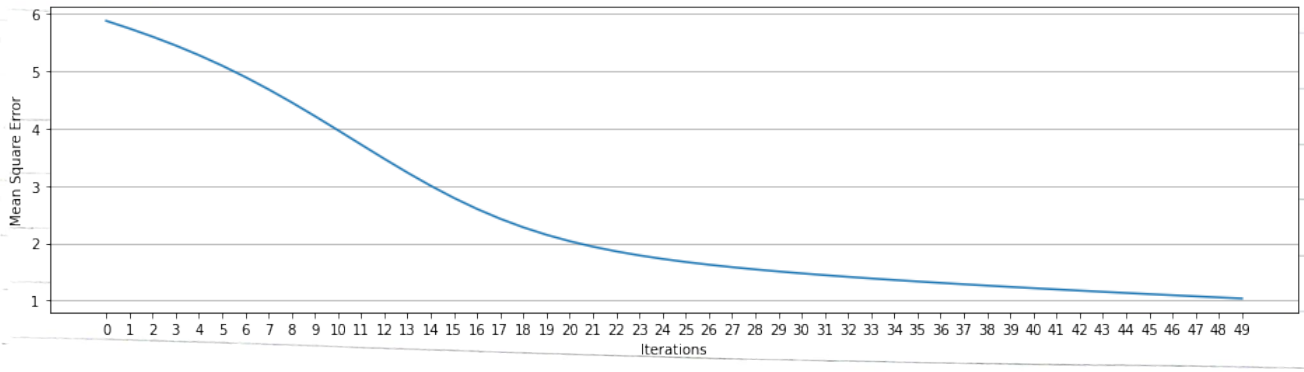
Q8 gamma=0.1



Q8 gamma=0.01



Q8 gamma=0.001



Q9 on next page

Q9 Let e_{ij} be an error component,

then, $e_{ij} = R_{ij}^2 + (p_i q_j)^2 - 2R_{ij}(p_i q_j)$

Now,

$$\text{Gradient} \quad \nabla(p_i q_j) = \begin{bmatrix} \frac{de_{ij}}{dp_i} \\ \frac{de_{ij}}{dq_j} \end{bmatrix} = \begin{bmatrix} 2p_i q_j^2 - 2R_{ij} q_j \\ 2q_j p_i^2 - 2R_{ij} p_i \end{bmatrix}$$

$$\text{Hessian} = \begin{bmatrix} \frac{d^2 e_{ij}}{dp_i^2} & \frac{d^2 e_{ij}}{dp_i dq_j} \\ \frac{d^2 e_{ij}}{dp_i dq_j} & \frac{d^2 e_{ij}}{dq_j^2} \end{bmatrix} = \begin{bmatrix} 2q_j^2 & 4p_i q_j - 2R_{ij} \\ 4p_i q_j - 2R_{ij} & 2p_i^2 \end{bmatrix}$$

if P_i and Q_j are '0' vectors,

then $H(0,0) = 2\begin{bmatrix} 0 & -R_{ij} \\ -R_{ij} & 0 \end{bmatrix}$

This is not Positive Semi Definite

So, using Second order Test of Convexity, we say that function is not convex

The function is differentiable, as it is a function in degree two.

Spiral

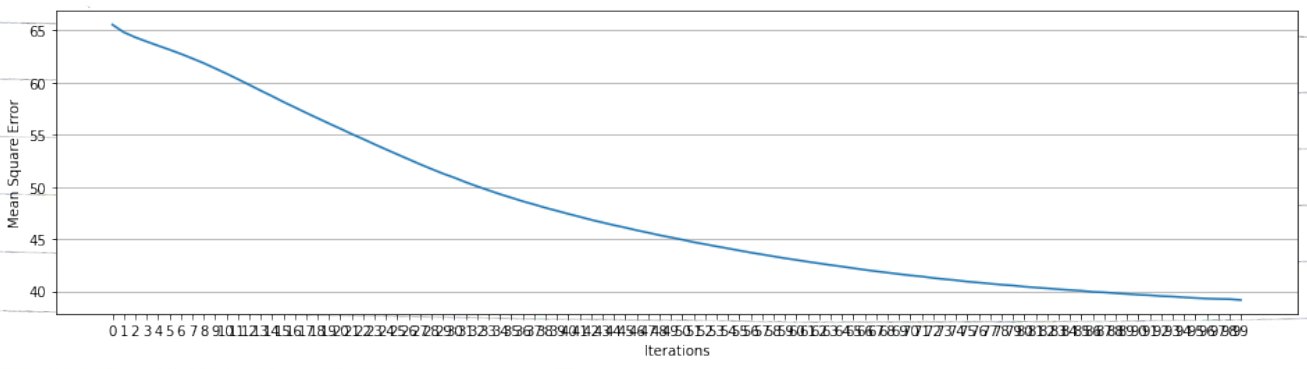
Question-10: In the reference paper [1] above, additional bias terms are recommended, implement it

Answer: Put your modified function here or below this cell.

n [ ]:
```
# These functions can be replaced with the functions present in the above class

    def predict(self, i, j):
            """
            Prediction: Predicted the rating of user i and item j
            """
            b_u=np.mean(self.R[i][np.where(self.R[i] != 0)])
            b_i=np.mean(self.R.T[j][np.where(self.R.T[j]!=0)])
            prediction = self.b + self.P[i, :].dot(self.Q[j, :].T)+(b_u-self.b)+(b_i-self.b)

            return prediction

    def full_matrix(self):
        """
        The rating matrix using the biases P and Q
        """
        new_values=np.zeros((self.R.shape[0],self.R.shape[1]))
        for i in range(R.shape[0]):
            for j in range(R.shape[1]):
                b_u=np.mean(self.R[i][np.where(self.R[i] != 0)])
                b_i=np.mean(self.R.T[j][np.where(self.R.T[j]!=0)])
                new_values[i][j]=self.b+(b_u-self.b)+(b_i-self.b)
        return new_values + self.P.dot(self.Q.T)
```

Here b_u is the user bias and b_i is the item bias , so, at the time of prediction i have added (b_u -self.b) and (b_i-self.b)

Q11 netflix graph for 100 iterations on combined_data_1.txt



Spiral

Code for Question 11

```
]: import numpy as np
with open('./combined_data_1.txt') as f:
    i=0
    arr=np.zeros((10000,1000))
    x=0
    while i<1000000:
        i+=1
        lines=f.readline()
        line=lines.split(',')
        if len(line)>1:
#             print(line[1])
            if (int)(line[0])<10000:
                arr[(int)(line[0])][x]=(int)(line[1])
        else:
            x+=1
    lf = LF(arr, K=2, alpha=0.01, gamma=0.001, iterations=100)
    training_process = lf.train()
    print()
    print("P x Q:")
    print(lf.full_matrix())
    print()
    print("Global bias:")
    print(lf.b)
    print()
    x = [x for x, y in training_process]
    y = [y for x, y in training_process]
    plt.figure(figsize=((16,4)))
    plt.plot(x, y)
    plt.xticks(x, x)
    plt.xlabel("Iterations")
    plt.ylabel("Mean Square Error")
    plt.grid(axis="y")
```