CS 7641 CSE/ISYE 6740 Homework 3 Solution

Rakesh Surapaneni

Nice and pleasent day of 11/13 Fri, 4 PM

1 Linear Regression [30 pts]

In class, we derived a closed form solution (normal equation) for linear regression problem: $\hat{\theta} = (X^T X)^{-1} X^T Y$. A probabilistic interpretation of linear regression tells us that we are relying on an assumption that each data point is actually sampled from a linear hyperplane, with some noise. The noise follows a zero-mean Gaussian distribution with constant variance. Specifically,

$$Y^i = \theta^T X^i + \epsilon^i \tag{1}$$

where $\epsilon^i \sim \mathcal{N}(0, \sigma^2 I)$, $\theta \in \mathbb{R}^d$, and $\{X^i, Y^i\}$ is the i-th data point. In other words, we are assuming that each every point is independent to each other and that every data point has same variance.

(a) Using the normal equation, and the model (Eqn. 1), derive the expectation $\mathbb{E}[\hat{\theta}]$. Note that here X is fixed, and only Y is random, i.e. "fixed design" as in statistics. [6 pts]

Solution:

- We know that $\hat{\theta} = (X^T X)^{-1} X^T Y$ (Note that each row of X is a data sample X^{i^T} in this notation for this equation to hold).
- Hence

$$\mathbb{E}[\hat{\theta}] = \mathbb{E}[(X^T X)^{-1} X^T Y]$$
$$= (X^T X)^{-1} X^T \mathbb{E}[Y]$$

  (since X is constant)

- We know that $Y^i = \theta^T X^i + \epsilon^i$. By stacking all Y's in a column, we can derive that $Y = X\theta + \epsilon^i$

$$= (X^T X)^{-1} X^T \mathbb{E}[X\theta + \epsilon^i]$$

  (By replacing Y)

$$= (X^T X)^{-1} X^T (\mathbb{E}[X\theta] + E[\epsilon^i])$$

- We know that noise follows a zero mean distribution. Hence $E[\epsilon^i] = 0$. Replacing the same thing in above equation we get:

$$= (X^T X)^{-1} X^T (\mathbb{E}[X\theta] + 0) = (X^T X)^{-1} X^T (\mathbb{E}[X\theta])$$

- Since X and θ are constants, we get $\mathbb{E}[X\theta] = X\theta$. Replacing the expectation in above equation we get,

$$= (X^T X)^{-1} X^T (X\theta) = (X^T X)^{-1} (X^T X)\theta = \theta$$

Hence,

$$\boxed{\mathbb{E}[\hat{\theta}] = \theta}$$

(b) Similarly, derive the variance Var$[\hat{\theta}]$. [6 pts]

Solution:

References for Matrix covariance property = $https://en.wikipedia.org/wiki/Covariance_matrix$

- We know that from above wiki property 3: $\text{Var}[AX + a] = A\text{Var}[X]A^T$ assuming A and a constant matrices/vectors.
- Using above property, we have

$$\text{Var}[\hat{\theta}] = \text{Var}[(X^TX)^{-1}X^TY] = (X^TX)^{-1}X^T\text{Var}[Y]((X^TX)^{-1}X^T)^T$$

- Assuming $X\theta$ is constant, Variance of Y equals variance of ϵ.Hence,

$$\text{Var}[\hat{\theta}] = (X^TX)^{-1}X^T\text{Var}[\epsilon]((X^TX)^{-1}X^T)^T$$

- Since ϵ is a zero mean distribution with Covariance $\Sigma = \sigma^2 I$

$$= (X^TX)^{-1}X^T\sigma^2 I((X^TX)^{-1}X^T)^T$$

$$= (X^TX)^{-1}X^T\sigma^2 IX(X^TX)^{-1} = \sigma^2(X^TX)^{-1}X^TX(X^TX)^{-1}$$

$$= \sigma^2(X^TX)^{-1}(X^TX)(X^TX)^{-1} = \sigma^2(X^TX)^{-1}$$

Hence,

$$\boxed{\text{Var}[\hat{\theta}] = \sigma^2(X^TX)^{-1}}$$

(c) Under the white noise assumption above, someone claims that $\hat{\theta}$ follows Gaussian distribution with mean and variance in (a) and (b), respectively. Do you agree with this claim? Why or why not? [8 pts]

Solution:

- Under white noise assumption, we know that ϵ follows Gaussian. Hence Y which is linearly dependent of ϵ follows a Gaussian distribution (kind of like a ridge around expected value).
- Now since Y follows a Gaussian distribution, $\hat{\theta}$ is a linear transformation on top of Y. ($\hat{\theta} = (X^TX)^{-1}X^TY$). Hence $\hat{\theta}$ follows Gaussian.
- The Mean and Variance is caclulated as per parts a and b as a dependent on that of ϵ. Hence they are valid values.
- furthur justification of mean can be provided. Now consider σ^2 be very low, infinitesimally small, then ϵ 0. In this scenario, since $\hat{\theta}$ is linearly related, Its value should be equal to mean which is θ since $X\hat{\theta} = X\theta$.
- Similarly variance can be justified since variance of $\hat{\theta}$ is directly proportional to variance of ϵ. This is justified since variance of estimate should increase with errors.

Therefore θ should follow Normal distribution with the mean and variance Matrix values justified and intuitive.

(d) Weighted linear regression

Suppose we keep the independence assumption but remove the same variance assumption. In other words, data points would be still sampled independently, but now they may have different variance σ_i. Thus, the covariance matrix of Y would be still diagonal, but with different values:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{bmatrix}. \tag{2}$$

Derive the estimator $\hat{\theta}$ (similar to the normal equations) for this problem using matrix-vector notations with Σ. [10 pts]

Solution:

- To simplify it let us scale X and Y by scaling each dimension(i) by $1/\sigma_i$. After this transformation, let P denote new X and Q denote Y (sclaed version).

  $$Y = X\theta + \epsilon$$

  let W denote weights we are trying to scale (W = $\Sigma^{-0.5}$)

  $$W = \begin{bmatrix} 1/\sigma_1 & 0 & \dots & 0 \\ 0 & 1/\sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma_n \end{bmatrix}$$

  By multiplying W to equation above, we get $WY = WX\theta + W\epsilon$

- note that since Variance matrix of $W\epsilon$ is I since $\text{Var}[ax] = a^2\text{Var}[x]$

- This looks similar to regular equation of excpet both X and Y are weighted. Solving for $\hat{\theta}$ should give us similar equation

  $$\hat{\theta} = ((WX)^T WX)^{-1}(WX)^T(WY)$$

  (Note that original equation itself is obtained by expectation maximization).

  $$\hat{\theta} = (X^T W^2 X)^{-1} X^T W^2 Y$$

  (Since $W^T = W$) By Replacing $W^2 by \Sigma^-1$

  $$\boxed{\hat{\theta} = (X^T \Sigma^{-1} X)^{-1} X^T \Sigma^{-1} Y}$$

- By doing similar calculations as in a, we get

  $$\mathbb{E}[\hat{\theta}] = \theta$$

- By doing similar calculations as in b, we get

  $$\text{Var}[\hat{\theta}] = ((WX)^T WX)^{-1}$$

  (Since new $\sigma = 1$, variance of scaled error is I).

  $$\text{Var}[\hat{\theta}] = (X^T W^2 X)^{-1} = (X^T \Sigma^{-1} X)^{-1} X^T W^2 Y$$

  (Since $\Sigma^-1 = W^2$)

Combining above two results we get that $\hat{\theta}$ follows the distribution

$$\boxed{\mathcal{N}(\theta, (X^T \Sigma^{-1} X)^{-1})} \tag{3}$$

2 Ridge Regression [15 pts]

For linear regression, it is often assumed that $y = \theta^\top \mathbf{x} + \epsilon$ where $\theta, \mathbf{x} \in \mathbb{R}^m$ by absorbing the constant term, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian random variable. Given n i.i.d samples $(\mathbf{x}^1, y^1), ..., (\mathbf{x}^n, y^n)$, we define $\mathbf{y} = (y^1, ..., y^n)^\top$ and $X = (\mathbf{x}^1, ..., \mathbf{x}^n)^\top$. Thus, we have $\mathbf{y} \sim \mathcal{N}(X\theta, \sigma^2 I)$. Show that the ridge regression estimate is the mean of the posterior distribution under a Gaussian prior $\theta \sim \mathcal{N}(0, \tau^2 I)$. Find the explicit relation between the regularization parameter λ in the ridge regression estimate of the parameter θ, and the variances σ^2, τ^2.

Solution:

- The posterior probability $P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)}$
- since P(y) is a constant and both the probability distributions are scaled Gaussians, we can simplify the equation as

$$P(\theta|y) = const * P(y|\theta)P(\theta)$$
$$P(\theta|y) = const * \mathcal{N}(X\theta, \sigma^2 I)\mathcal{N}(0, \tau^2 I)$$
$$P(\theta|y) = const * e^{-\frac{||y-X\theta||^2}{2\sigma^2}} e^{-\frac{||\theta||^2}{2\tau^2}}$$

$$P(\theta|y) = const * e^{-\frac{||y-X\theta||^2}{2\sigma^2} - \frac{||\theta||^2}{2\tau^2}}$$
$$P(\theta|y) = const * e^{-\frac{||y-X\theta||^2 + (\sigma^2/\tau^2)||\theta||^2}{2\sigma^2}}$$

- The above equation can be rephrased into a Gaussian format by rearranging terms and taking residue as a constant term. Hence the prior is a Gaussian form.

$$P(\theta|y) = const * e^{-\frac{(y-X\theta)^T(y-X\theta) + (\sigma^2/\tau^2)\theta^T\theta}{2\sigma^2}}$$

$$P(\theta|y) = const * e^{-\frac{(y^T - \theta^T X^T)(y-X\theta) + (\sigma^2/\tau^2)\theta^T\theta}{2\sigma^2}}$$
$$= const * e^{-\frac{(\theta^T(X^TX)\theta) + (\sigma^2/\tau^2)\theta^T\theta + y^Ty - y^TX\theta - (X\theta)^Ty}{2\sigma^2}}$$

- Let us call σ^2/τ^2 as λ. The above equation reduces to

$$= const * e^{-\frac{(\theta^T(X^TX+\lambda I)\theta) + y^Ty - y^TX\theta - (\theta)^TX^Ty}{2\sigma^2}}$$
$$= const * e^{-\frac{(\theta^T(X^TX+\lambda I)\theta) + y^Ty - y^TX(X^TX+\lambda I)^{-1}(X^TX+\lambda I)\theta - (\theta)^T(X^TX+\lambda I)(X^TX+\lambda I)^{-1}X^Ty}{2\sigma^2}}$$
$$= const * e^{-\frac{(\theta-\mu)^T\Sigma^{-1}(\theta-\mu)}{2} + const}$$

  Where $\Sigma = (X^TX + \sigma^2/\tau^2 I)$ and $\mu = (X^TX + \sigma^2/\tau^2 I)^{-1}X^Ty$

- Hence posterior distribution is Gaussian. with mean $\boxed{\mu = (X^TX + \sigma^2/\tau^2 I)^{-1}X^Ty}$
- This is of the form of ridge regression solution $\theta = (X^TX + \lambda I)^{-1}X^Ty$
- the regularization parameter is ratio of both the variances. $\boxed{\lambda = \frac{\sigma^2}{\tau^2}}$

3 Bias - Variance Decomposition [15 pts]

Suppose x is a d-dimensional vector. The estimator S^2 defined as

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_j^i - \bar{\mathbf{x}}_j)^2$$

where $j = 1, 2, ..., d$ and $\bar{\mathbf{x}}_j = \frac{1}{n}\sum \mathbf{x}_j$, is used to estimate the diagonal of covariance matrix, that is, $diag(Cov(X))$. Show that S^2 is an unbiased estimator.

Solution:

- By calculating the value S^2, we are trying to estimate trace of co-variance of the distribution of random variable x, a d-dimensional vector vector.
- We sample x, n times and i'th sample is represented by x^i.
- To prove that S^2 is an unbiased estimator, we calculate expected value of S^2 and prove that it is indeed trace of variance matrix.

$$\mathbb{E}[S^2] = \mathbb{E}[\frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_j^i - \bar{\mathbf{x}}_j)^2]$$

- to make calculations simple, we try to eliminate mean from the calculations. Let μ be mean of distribution of x.
  Let $y^i = x^i - \mu$

$$\mathbb{E}[S^2] = \mathbb{E}[\frac{1}{n-1}\sum_{i=1}^{n}((\mathbf{x}_j^i - \mu) - (\bar{\mathbf{x}}_j - \mu))^2]$$

$$= \mathbb{E}[\frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{y}_j^i - \bar{\mathbf{y}}_j)^2]$$

$$= \mathbb{E}[\frac{1}{n-1}\sum_{i=1}^{n}((\mathbf{y}_j^i)^2 + (\bar{\mathbf{y}}_j)^2 - 2\mathbf{y}_j^i\bar{\mathbf{y}}_j)]$$

- Let d denotes the sum of diagonal of co-variance matrix for x, which we are trying to calculate. Note that co-variance matrix doesn't change by shifting mean and hence co-variance matrix remains same.

$$d = \mathbb{E}[(x-\mu)^2] = \mathbb{E}[y^2]$$

$$=> \mathbb{E}[S^2] = \mathbb{E}[\frac{1}{n-1}\sum_{i=1}^{n}((\mathbf{y}_j^i)^2 + (\bar{\mathbf{y}}_j)^2 - 2\mathbf{y}_j^i\bar{\mathbf{y}}_j)]$$

$$= \frac{1}{n-1}\sum_{i=1}^{n}(\mathbb{E}[(\mathbf{y}_j^i)^2] + \mathbb{E}[(\bar{\mathbf{y}}_j)^2] - 2\mathbb{E}[\mathbf{y}_j^i\bar{\mathbf{y}}_j])$$

$$= \frac{1}{n-1}\sum_{i=1}^{n}(d + \mathbb{E}[(\bar{\mathbf{y}}_j)^2] - 2\mathbb{E}[\mathbf{y}_j^i\bar{\mathbf{y}}_j])$$

$$\mathbb{E}[(\bar{\mathbf{y}}_j)^2] = \mathbb{E}[(\frac{1}{n}\sum \mathbf{y}^j)^2] = \frac{1}{n^2}\mathbb{E}[(\sum(\mathbf{y}^j)^2 + 2\sum_{i!=j}\mathbf{y}^j\mathbf{y}^i)]$$

- Assuming independent distribution, we know that $\mathbb{E}[\sum_{i!=j}\mathbf{y}^j\mathbf{y}^i] = \sum_{i!=j}\mathbb{E}[\mathbf{y}^j]\mathbb{E}[\mathbf{y}^i] = 0$

- Hence
$$\mathbb{E}[(\bar{\mathbf{y}}_j)^2] = \frac{1}{n^2}\mathbb{E}[(\sum(\mathbf{y}^j)^2)] = \frac{1}{n^2}nd = \frac{1}{n}d$$
- for the last part of equation
$$\mathbb{E}[\mathbf{y}_j^i\bar{\mathbf{y}}_j] = 1/n * \sum \mathbb{E}[\mathbf{y}^j * \mathbf{y}^i]$$
- Using $\mathbb{E}[\mathbf{y}^j * \mathbf{y}^i] = 0$ for all $i! = j$, we get
$$\mathbb{E}[\mathbf{y}_j^i\bar{\mathbf{y}}_j] = 1/n * \sum \mathbb{E}[{\mathbf{y}_j^i}^2] = \frac{1}{n}d$$
- putting above results in the equations for $\mathbb{E}[s^2]$, we get
$$=> \mathbb{E}[S^2] = \frac{1}{n-1}\sum_{i=1}^{n}(d + \mathbb{E}[(\bar{\mathbf{y}}_j)^2] - 2\mathbb{E}[\mathbf{y}_j^i\bar{\mathbf{y}}_j]) = \frac{1}{n-1}\sum_{i=1}^{n}(d + \frac{d}{n} - 2\frac{d}{n})$$
$$= \frac{1}{n-1}\sum_{i=1}^{n}(d - \frac{d}{n}) = \frac{1}{n-1}\sum_{i=1}^{n}(d * \frac{n-1}{n} = d$$

Hence Expected value of S^2 is the sum of diagonal of co-variance matrix of X.
Therefore S^2 is an unbiased estimator.

4 Programming: Recommendation System [40 pts]

Personalized recommendation systems are used in a wide variety of applications such as electronic commerce, social networks, web search, and more. Machine learning techniques play a key role to extract individual preference over items. In this assignment, we explore this popular business application of machine learning, by implementing a simple matrix-factorization-based recommender using gradient descent.

Suppose you are an employee in Netflix. You are given a set of ratings (from one star to five stars) from users on many movies they have seen. Using this information, your job is implementing a personalized rating predictor for a given user on unseen movies. That is, a rating predictor can be seen as a function $f : \mathcal{U} \times \mathcal{I} \to \mathbb{R}$, where $\mathcal{U}$ and $\mathcal{I}$ are the set of users and items, respectively. Typically the range of this function is restricted to between 1 and 5 (stars), which is the the allowed range of the input.

Now, let's think about the data representation. Suppose we have m users and n items, and a rating given by a user on a movie. We can represent this information as a form of matrix, namely rating matrix M. Suppose rows of M represent users, while columns do movies. Then, the size of matrix will be $m \times n$. Each cell of the matrix may contain a rating on a movie by a user. In $M_{15,47}$, for example, it may contain a rating on the item 47 by user 15. If he gave 4 stars, $M_{15,47} = 4$. However, as it is almost impossible for everyone to watch large portion of movies in the market, this rating matrix should be very sparse in nature. Typically, only 1% of the cells in the rating matrix are observed in average. All other 99% are missing values, which means the corresponding user did not see (or just did not provide the rating for) the corresponding movie. Our goal with the rating predictor is estimating those missing values, reflecting the user's preference learned from available ratings.

Our approach for this problem is matrix factorization. Specifically, we assume that the rating matrix M is a low-rank matrix. Intuitively, this reflects our assumption that there is only a small number of factors (e.g, genre, director, main actor/actress, released year, etc.) that determine like or dislike. Let's define r as the number of factors. Then, we learn a user profile $U \in \mathbb{R}^{m \times r}$ and an item profile $V \in \mathbb{R}^{n \times r}$. (Recall that m and n are the number of users and items, respectively.) We want to approximate a rating by an inner product of two length r vectors, one representing user profile and the other item profile. Mathematically, a rating by user u on movie i is approximated by

$$M_{u,i} \approx \sum_{k=1}^{r} U_{u,k} V_{i,k}. \tag{4}$$

We want to fit each element of U and V by minimizing squared reconstruction error over all training data points. That is, the objective function we minimize is given by

$$E(U, V) = \sum_{(u,i) \in M} (M_{u,i} - U_u^T V_i)^2 = \sum_{(u,i) \in M} (M_{u,i} - \sum_{k=1}^{r} U_{u,k} V_{i,k})^2 \tag{5}$$

where U_u is the uth row of U and V_i is the ith row of V. We observe that this looks very similar to the linear regression. Recall that we minimize in linear regression:

$$E(\theta) = \sum_{i=1}^{m} (Y^i - \theta^T x^i)^2 = \sum_{i=1}^{m} (Y^i - \sum_{k=1}^{r} \theta_k x_k^i)^2 \tag{6}$$

where m is the number of training data points. Let's compare (5) and (6). $M_{u,i}$ in (5) corresponds to Y^i in (6), in that both are the observed labels. $U_u^T V_i$ in (5) corresponds to $\theta^T x^i$ in (6), in that both are our estimation with our model. The only difference is that both U and V are the parameters to be learned in (5), while only θ is learned in (6). This is where we personalize our estimation: with linear regression, we apply the same θ to any input x^i, but with matrix factorization, a different profile U_u are applied depending on who is the user u.

As U and V are interrelated in (5), there is no closed form solution, unlike linear regression case. Thus, we need to use gradient descent:

$$U_{v,k} \leftarrow U_{v,k} - \mu \frac{\partial E(U,V)}{\partial U_{v,k}}, \qquad V_{j,k} \leftarrow V_{j,k} - \mu \frac{\partial E(U,V)}{\partial V_{j,k}}, \tag{7}$$

where μ is a hyper-parameter deciding the update rate. It would be straightforward to take partial derivatives of $E(U,V)$ in (5) with respect to each element $U_{v,k}$ and $V_{j,k}$. Then, we update each element of U and V using the gradient descent formula in (7).

(a) Derive the update formula in (7) by solving the partial derivatives. [10 pts]

Solution
from (5)

$$E(U,V) = \sum_{(u,i)\in M} (M_{u,i} - \sum_{k=1}^{r} U_{u,k} V_{i,k})^2$$

$$\frac{\partial E(U,V)}{\partial U_{v,k}} = - \sum_{(v,i)\in M} 2(M_{v,i} - \sum_{p=1}^{r} U_{v,p} V_{i,p}) V_{i,k}$$

(by applying chain rule) Similarly

$$\frac{\partial E(U,V)}{\partial V_{j,k}} = - \sum_{(u,j)\in M} 2(M_{u,j} - \sum_{p=1}^{r} U_{u,p} V_{j,p}) V_{j,k}$$

BY substituting the above value in (7), we get

$$U_{v,k} \leftarrow U_{v,k} - \mu \frac{\partial E(U,V)}{\partial U_{v,k}} = U_{v,k} - 2\mu \sum_{(v,i) in M} V_{i,k}(M_{v,i} - \sum_{p=1}^{r} U_{v,p} V_{i,p})$$

$$V_{j,k} \leftarrow U_{j,k} - \mu \frac{\partial E(U,V)}{\partial V_{j,k}} = U_{u,k} - 2\mu \sum_{(u,j) in M} V_{j,k}(M_{u,j} - \sum_{p=1}^{r} U_{u,p} V_{j,p})$$

(b) To avoid overfitting, we usually add regularization terms, which penalize for large values in U and V. Redo part (a) using the regularized objective function below. [5 pts]

$$E(U,V) = \sum_{(u,i)\in M} (M_{u,i} - \sum_{k=1}^{r} U_{u,k} V_{i,k})^2 + \lambda \sum_{u,k} U_{u,k}^2 + \lambda \sum_{i,k} V_{i,k}^2$$

(λ is a hyper-parameter controlling the degree of penalization.)
Solution

$$E(U,V) = \sum_{(u,i)\in M} (M_{u,i} - \sum_{k=1}^{r} U_{u,k} V_{i,k})^2 + \lambda \sum_{u,k} U_{u,k}^2 + \lambda \sum_{i,k} V_{i,k}^2$$

$$\frac{\partial E(U,V)}{\partial U_{v,k}} = - \sum_{(v,i)\in M} 2(M_{v,i} - \sum_{p=1}^{r} U_{v,p} V_{i,p}) V_{i,k} + 2\lambda U_{v,k}$$

$$\frac{\partial E(U,V)}{\partial V_{j,k}} = - \sum_{(u,j)\in M} 2(M_{u,j} - \sum_{p=1}^{r} U_{u,p} V_{j,p}) V_{j,k} + 2\lambda V_{j,k}$$

Hence the gradient update formula is

$$\boxed{U_{v,k} \leftarrow (1-2\lambda)U_{v,k} - 2\mu \sum_{(v,i) in M} V_{i,k}(M_{v,i} - \sum_{p=1}^{r} U_{v,p}V_{i,p})}$$

$$\boxed{V_{j,k} \leftarrow (1-2\lambda)U_{u,k} - 2\mu \sum_{(u,j) in M} V_{j,k}(M_{u,j} - \sum_{p=1}^{r} U_{u,p}V_{j,p})}$$

(c) Implement `myRecommender.m` by filling the gradient descent part.

Evaluation [15 pts]

Upload your `myRecommender.m` implementation file. (Do not copy and paste your code in your report. Be sure to upload your `myRecommender.m` file.)

Submitted the code separately.

Report [10 pts]

In your report, show the performance (RMSE) both on your training set and test set, with varied `lowRank`. (The default is set to 1, 3, and 5, but you may want to vary it further.) Discuss what you observe with varied low rank. Also, briefly discuss how you decided your hyper-parameters (μ, λ).

Report

- **Selection of μ, the update rate :** The selection of μ seems to affect the convergence of the gradient descent algorithm. Only for a narrow band of values, the convergence is observed
  Following is the observation for 100 iterations, r = 5, $\lambda = 0$

μ	converging?	train RMSE	test RMSE
>= 0.1	No	NaN	NaN
>= 0.01	No	NaN	NaN
>= 0.005	No	NaN	NaN
0.0001	Yes	1.0454	1.0760
0.00001	Yes	3.3798	3.3704
0.0005	Yes	0.9062	0.9489
0.00055	Yes	0.9011	0.9504
0.000584	Yes	0.8821	0.9403

  0.000584 seems to be performing well for these parameters and Would like to assign this value for further testing.

- **Selection of μ, the regularization rate :** Following is the observation for 100 iterations, r = 5, μ = 0.000584

λ	train RMSE	test RMSE
100	1.6342	1.6518
10	0.9517	0.9874
1	0.8970	0.9452
0.1	0.8899	0.9398
0.01	0.8927	0.9460
0.5	0.8808	0.9393
0.4	0.8843	0.9379

λ value of 0.4 seems to perform decently and hence is chosen value.

- **Effect of lowrank:**

low rank	train RMSE	test RMSE	time
1	0.9190	0.9484	3.42
3	0.8998	0.9504	4.21
5	0.8573	0.9415	5.62
7	0.8635	0.9452	6.13
9	0.8699	0.9427	6.42
11	0.8517	0.9383	5.60
13	0.8437	0.9370	6.89
15	0.8576	0.9450	4.94
17	0.8634	0.9472	5.42
20	0.8675	0.9420	4.36

- Experiment removing convergence condition and allowing loops through run through gave better results

low rank	train RMSE	test RMSE	time
1	0.9707	1.0137	71.94
3	0.8319	0.9381	81.56
5	0.7738	0.9716	75.06
7	0.7248	1.0156	74.30
9	0.6808	1.0587	71.31
11	0.6386	1.0939	72.44
13	0.6039	1.1178	82.13

- With increased rank, discrepency between train RMS and test RMSE is increasing. Clearly for very high rank ¿ 10, with enough iterations, we see overfitting where test errors are getting worse while train errors are very minimal. Hence lowrank = 5 is the best value.