1. for basic objective function $\mathrm{E(u,v)} = \sum_{u,i\epsilon O}\left(M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}\right)^2$

Consider a single term of E(U,V),

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

$$\frac{\partial E_{u,i}(U,V)}{\partial U_{u,k}} = 2\left(M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}\right)\left(-V_{i,k}\right) = 2V_{i,k}\left(\sum_{k=1}^{r} U_{u,k}V_{i,k} - M_{u,i}\right) = -2V_{i,k}R_{u,i}$$

Here $R_{u,i}$ is current error, $R_{u,i} = M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}$,

Similarly, $\frac{\partial E_{u,i}(U,V)}{\partial V_{i,k}} = -2U_{u,k}R_{u,i}$

Update equation: $U_{u,k} \leftarrow U_{u,k} + 2\mu V_{i,k}R_{u,i} \quad V_{i,k} \leftarrow V_{i,k} + 2\mu U_{u,k}R_{u,i}$, for each (u,i)$\epsilon O$

2. for regularized objective function: $\mathrm{E(u,v)} = \sum_{u,i\epsilon O}\left(M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}\right)^2 + \lambda(\sum_{u,k} U_{u,k}^2 + \sum_{i,k} V_{i,k}^2)$

Also consider a single term firstly, and the derivatives just plug in a penalize part, $R_{u,i}$ is current error, $R_{u,i} = M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}$

$$\frac{\partial E_{u,i}(U,V)}{\partial U_{u,k}} = -2V_{i,k}R_{u,i} + 2\,\lambda U_{u,k}$$

$$\frac{\partial E_{u,i}(U,V)}{\partial V_{i,k}} = -2U_{u,k}R_{u,i} + 2\,\lambda V_{i,k}$$

$U_{u,k} \leftarrow U_{u,k} + 2\mu(V_{i,k}R_{u,i} - \lambda U_{u,k}), \quad V_{i,k} \leftarrow V_{i,k} + 2\mu\left(U_{u,k}R_{u,i} - \lambda V_{i,k}\right)$ for each (u,i)ϵO

3.

RMSE	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$
$\mu = 0.0001$	1.0366	1.0620	1.1894
$\mu = 0.0005$	0.9818	0.9881	1.0949
$\mu = 0.001$	0.9753	0.9733	1.0863

Table 1: RMSE for r = 1

RMSE	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$
$\mu = 0.0001$	1.2778	1.2615	1.4065
$\mu = 0.0005$	1.0067	1.0173	1.1096
$\mu = 0.001$	0.9938	0.9874	1.0903

Table 2: RMSE for r = 3

RMSE	$\lambda = 0.05$	$\lambda = 0.1$	$\lambda = 0.5$
$\mu = 0.0001$	1.5889	1.4330	1.2380
$\mu = 0.0005$	1.0298	1.0377	1.1243
$\mu = 0.001$	1.0054	1.0031	1.1060

Table 3: RMSE for r = 5

μ	0.001
λ	0.1
r	1
RMSE	0.9733

Table 4: The Best Model

1) What do you observe when you vary r? Why?

Under the same configuration set of μ and λ, when increasing r, the rmse will increase. Simply to say, we have projected users and items onto r-dimensional space. From information theory, the number of instance needed will increase exponentially with the increase of dimensions. In this circumstance, the size training data set is fixed, to deal with higher dimensions, rmse increased, namely, performance decreased.

Additionally, the running time also increase because the computational cost is O(|(u,i)|*r).

2) Which model is the best? Please describe the best model in Table 4 and explain your choice..

The best model is under the configuration: learning rate 0.001, regularization parameter 0.1, the number of latent factor 1 because the rmse on test data under 10-fold cross-validation is the smallest one in this experiment. As for learning rate, if set it too low, it is likely to get stuck with local minima. If choose λ too small, it's likely to overfitting; if choose λ too large, it's likely to underfitting. Thus, with the validation set method, we could find an optimal parameter configuration.

3) Suppose you are using regularized MF in real systems, how will you choose parameters? Why?

First of all, because the storage cost is O(|O|+(M+N)*r), here M*r, N*r is the dimensions for user and item profile, respectively, when dealing with extreme large rating records, which is in real systems, it's likely to out of core memory. Parallel implementation could be considered to address this problem but needs more complex setting.

Then, since gradient descent is a local search method, it suffers from being stuck with local minima issue. The learning rate should be fixed under multiple experiments with the object data as well as the stop criteria.

Additionally, because the regularized parameter will be influenced by iteration, and it impact whether the model is overfitting or underfitting, we should do experiments with cross validation.

The withheld validation set is useful for choice of parameters, since the predict accuracy for unseen data can reflect the performance of the algorithm.