As a reminder, for users $u \in R^d$ and movies $v \in R^d$, we have $u_i \sim N(0,\lambda^{-1}I)$, $i = 1,\ldots,N_1$, $v_j \sim N(0,\lambda^{-1}I)$, $j = 1,\ldots,N_2$.

We are given an $N_1 \times N_2$ matrix M with missing values. Given the set $\Omega = \{(i, j) : M_{ij}$ is measured$\}$, for each $(i,j) \in \Omega$ we model $M_{ij} \sim N(u^T_i v_j,\sigma^2)$.

For the algorithm, set $\sigma^2 = 0.25$, $d = 10$ and $\lambda = 1$. Train the model on the larger training set for 100 iterations. For each user-movie pair in the test set, predict the rating using the relevant dot product. Note that the mean rating has been subtracted from the data and you do not need to round your prediction. Since the equations are in the slides, there's no need to re-derive it.

. a) Run your code 10 times. For each run, initialize your u_i and v_j vectors as $N(0, I)$ random vectors. On a *single* plot, show the the log joint likelihood for iterations 2 to 100 for each run. In a table, show in each row the final value of the training objective function next to the RMSE on the testing set. Sort these rows according to decreasing value of the objective function.

. b) For the run with the highest objective value, pick the movies "Star Wars" "My Fair Lady" and "Goodfellas" and for each movie find the 10 closest movies according to Euclidean distance using their respective locations v_j . List the query movie, the ten nearest movies and their distances. A mapping from index to movie is provided with the data.