COMS 4721 – Machine Learning for Data Science

Daniel Kost-Stephenson – dpk2124

April 21st, 2016

Question 1

Part a)

The MLE estimator for b_i holding all other variables constant can be found by taking the derivative of the log likelihood function with respect to b_i and setting the function to zero. The log likelihood function is given by:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{A}) \;:=\; -\frac{1}{2} \sum_{(i,j)\in\Omega} \left(\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle + b_i + c_j + \mu - a_{i,j}\right)^2 .$$

By taking the derivative and setting it equal to zero, we obtain:

$$\frac{\partial \boldsymbol{\mathcal{L}}(\boldsymbol{\theta}; \boldsymbol{\mathcal{A}})}{\partial b_i} = -\sum_{i,j} \left(\langle \boldsymbol{u_i}, \boldsymbol{v_j} \rangle + \widehat{b_\iota} + c_j + \mu - a_{i,j}\right) = 0$$

$$\widehat{b_\iota} = \frac{1}{n} \sum_{j}^{n} \left(a_{i,j} - c_j - \mu - \langle \boldsymbol{u_i}, \boldsymbol{v_j} \rangle\right) \; for\; i = 1, \dots, m$$

By using the same method, we can derive an expression for c_j:

$$\widehat{c_\jmath} = \frac{1}{m} \sum_{i}^{m} \left(a_{i,j} - b_i - \mu - \langle \boldsymbol{u_i}, \boldsymbol{v_j} \rangle\right) \; for\; j = 1, \dots, n$$

Expressions for u_i and v_j are also required for the alternating least squares algorithm. By taking the derivative of the log likelihood function and setting it to zero, we can obtain closed form equations that maximize these variables.

$$\frac{\partial \boldsymbol{\mathcal{L}}(\boldsymbol{\theta}; \boldsymbol{\mathcal{A}})}{\partial u_i} = -\sum_{i,j} \left(\langle \widehat{\boldsymbol{u}_\iota}, \boldsymbol{v_j} \rangle + b_i + c_j + \mu - a_{i,j}\right) \cdot \boldsymbol{v_j} = 0$$

$$-\sum_{i,j} \langle \widehat{\boldsymbol{u}_\iota}, \boldsymbol{v_j} \rangle (\boldsymbol{v_j}) - \sum_{i,j} (b_i + c_j + \mu - a_{i,j})(\boldsymbol{v_j}) = 0$$

$$\left(\sum_{i,j} \boldsymbol{v_j} \boldsymbol{v_j^T}\right) \widehat{u_\iota} = -\sum_{i,j} (b_i + c_j + \mu - a_{i,j})(\boldsymbol{v_j})$$

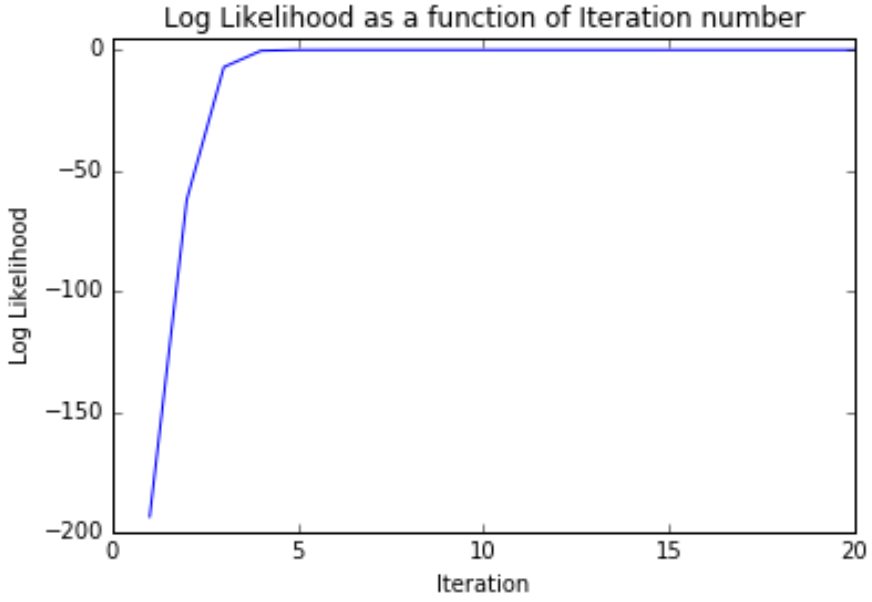
$$\widehat{u_i} = \left(-\sum_j^n (b_i + c_j + \mu - a_{i,j})(\boldsymbol{v_j}) \right) \cdot \left(\sum_j^n \boldsymbol{v_j v_j^T} \right)^{-1} \quad for\ i = 1, \dots, m$$

The expression for v_j follows as:

$$\widehat{v_j} = \left(-\sum_i^m (b_i + c_j + \mu - a_{i,j})(\boldsymbol{u_i}) \right) \cdot \left(\sum_i^m \boldsymbol{u_i u_i^T} \right)^{-1} \quad for\ j = 1, \dots, n$$

Part b)

The log likelihood function converges to zero when $T = 20$. The plot of the log likelihood as a function of number of iterations is show below:



Part c)

The problem that encountered is that the $\left(\sum_i^m \boldsymbol{u_i u_i^T}\right)^{-1}$ matrix in the update v_j step is a singular matrix, meaning it is not invertible. When that matrix is not invertible, there is no solution to the alternating least squares algorithm. The reason why the above matrix is not always invertible is because some movies have no user ratings, so python attempts to take the inverse of an element with nothing in it and raises an error.

Part d)

With the regularization term added, the $\left(\sum_i^m \boldsymbol{u_i u_i^T}\right)^{-1}$ matrix is *always* invertible, so the alternating least squares algorithm is non-decreasing with each iteration. The plot can be viewed below: although the log likelihood function converges to -32500 (which seems fairly large), the test RMSE is constantly decreasing and levels out at around 0.93. Given the context, this seems like a reasonable result: the

predicted rating for a given movie and user is off by less than one. The plots for RMSE as a function of iteration number and log likelihood as a function of iteration number are show below.