CS 363D/ SSC 358	Statistical Learning and Data Mining	Spring 2014

Homework 3

Lecturer: Pradeep Ravikumar — Date Due: Apr 07, 2014

Keywords: *Netflix, Regression, Least Squares*

Note Use the following command to turn in your code:

```
$ turnin --submit adarsh hw3 <filenames>
```

Please submit a hard-copy of just the results in class.

The Netflix Problem For this homework, you will work on the Netflix dataset, to predict missing (movie, user) ratings. The problem of predicting missing values for a recommender system is formally known as Collaborative Filtering. Read more about the Netflix challenge and the dataset at: `http://www.netflixprize.com`. The complete Netflix dataset has 480,189 users, 17,770 movies, and 100,480,507 ratings, so that the scale of the problem is huge. For this homework, we will work on a much smaller subset of 1978 users, 4635 movies and 166,749 ratings. You can download a zip file containing the dataset and a README file from Piazza.

Matrix Factorization We will use the Matrix Factorization approach described below. Let there be u users and m movies and let $R \in \mathbb{R}^{u \times m}$ be the ratings matrix where element R_{ij} is the rating given by user i for movie j. In the Matrix Factorization method, R is approximated by UM^T, where $U \in \mathbb{R}^{u \times k}$ and $M \in \mathbb{R}^{m \times k}$ and $k \ll \min(u, m)$. Intuitively, you can think of row $\boldsymbol{u}_i^T$ of U as a low-dimensional feature vector for the i-th user, and the row $\boldsymbol{m}_j^T$ of M as a low-dimensional feature vector for the j-th movie. Using the above notation, the rating R_{ij} for a movie j by user i is approximated as $R_{ij} \approx \boldsymbol{u}_i^T \boldsymbol{m}_j$. Note that the matrix R has many missing values and hence the problem can be stated formally as:

$$\min_{U,M} \ \frac{1}{2} \sum_{(i,j) \in \mathcal{K}} (R_{ij} - \boldsymbol{u}_i^T \boldsymbol{m}_j)^2 + \frac{1}{2}\lambda (\sum_{i=1}^{u} \|\boldsymbol{u}_i\|_2^2 + \sum_{j=1}^{m} \|\boldsymbol{m}_j\|_2^2), \tag{1}$$

where $\mathcal{K} = \{(i,j) \mid R_{ij} \text{ is known}\}$ and $\|\cdot\|_2$ denotes the ℓ_2 norm. In the above formulation, we are using **ridge regularization**, with regularization parameter $\lambda \geq 0$.

Alternating Minimization We will use the method of Alternating Minimization discussed in class to solve for both U and M, where in each iteration we fix U to compute M, and subsequently fix M to compute U, and so on. First, consider computing M while assuming U to be fixed. Let $\mathcal{K}_j$ denote the set of known ratings for movie j and let there be n_j known ratings for movie j; then (1) reduces to:

$$\min_{M = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_m]^T} \frac{1}{2} \sum_{j=1}^{m} \|\boldsymbol{r}_j^{(\mathcal{K}_j)} - U^{(\mathcal{K}_j)} \boldsymbol{m}_j\|_2^2 + \frac{1}{2}\lambda \sum_{j=1}^{m} \|\boldsymbol{m}_j\|_2^2, \tag{2}$$

where $\boldsymbol{r}_j^{\mathcal{K}_j} \in \mathbb{R}^{n_j}$ denotes the n_j known ratings for movie j and $U^{(\mathcal{K}_j)} \in \mathbb{R}^{n_j \times k}$ denotes the sub matrix of U for the corresponding users who rated movie j. Thus, we can solve a separate ridge regression problem for each j, i.e.,

$$\min_{\boldsymbol{m}_j} \ \|\boldsymbol{r}_j^{(\mathcal{K}_j)} - U^{(\mathcal{K}_j)} \boldsymbol{m}_j\|_2^2 + \lambda \|\boldsymbol{m}_j\|_2^2. \tag{3}$$

As we have learned in class, the solution to the above problem for movie j is

$$\boldsymbol{m}_j = (U^{(\mathcal{K}_j)^T} U^{(\mathcal{K}_j)} + \lambda I)^{-1} U^{(\mathcal{K}_j)^T} \boldsymbol{r}_j^{(\mathcal{K}_j)}, \tag{4}$$

where I is a $k \times k$ identity matrix. Similarly, let M be fixed, and let $\mathcal{K}_i$ denote the set of known ratings for user i and there be n_i known ratings by user i. The solution for user i is

$$\boldsymbol{u}_i = (M^{(\mathcal{K}_i)^T} M^{(\mathcal{K}_i)} + \lambda I)^{-1} M^{(\mathcal{K}_i)^T} \boldsymbol{r}_i^{(\mathcal{K}_i)}, \tag{5}$$

where $\boldsymbol{r}_i^{(\mathcal{K}_i)} \in \mathbb{R}^{n_i}$ denotes the n_i known ratings by user i and $M^{(\mathcal{K}_i)} \in \mathbb{R}^{n_i \times k}$ denotes the sub-matrix of M for the corresponding movies rated by user i.

So, for each iteration, compute U and then M using the procedure given above. Repeat this procedure for a fixed number of iterations. Initialize U and M randomly, fix the number of iterations τ as 30 and fix $k = 10$.

1. (11 points) Implement the method described above in MATLAB. Test your code using ten-fold cross validation (validation sets are provided in the .mat file) for different values of the regularization parameter λ (restricted to the 21 values $\{0, 0.05, 0.1, ...1\}$); and plot Root Mean Square Error (RMSE) averaged over all the validation sets vs λ.

2. (3 points) Report the optimal λ.

3. (3 points) When $\lambda = 0$, what problems do you face?

4. (3 points) For the optimal λ, what is the RMSE obtained for the test set?