Internet Analytics (COM-308)

Homework Set 5

Exercise 1

The rating matrix of five users for four items with few missing entries is given by

$$R = \begin{bmatrix} 2 & 1 & 3 & - \\ 3 & 2 & 5 & 5 \\ 5 & - & 4 & 2 \\ 4 & 3 & - & 4 \\ - & 1 & 5 & 3 \end{bmatrix}.$$

(a) First we try a simple predictor that takes into account only user bias. Compute the optimal baseline predictor $\overline{r} + b_u$ (without regularization) and the RMSE for this predictor. Feel free to rely on a tool such as matlab, octave, or mathpy to solve the resulting system of equations.

We need to solve $\min_{b_u} \sum_{(u,i)\in R}(r_{ui} - \overline{r} - b_u)^2$, where $\overline{r} = 3.25$. We first transform this cost function into the standard form $||Ab - c||^2$, as follows. Each row $A_{k.}$ of A corresponds to one (u, i) pair, and has a one at position u. The vector b is $[b_u]^T$, and c is $[r_k - \overline{r}]^T$.

The matrix $A^T A$ is rank complete with rank $r = 5$, so the minimum is unique. We find $b_u = [-1.25, 0.5, 0.42, 0.42, -0.25]$. Verify that your root mean squared error is 1.1748.

(b) Now we refine this predictor to take into account both a user and an item bias, as seen in class. Compute the optimal baseline predictor $\overline{r} + b_u + b_i$ (without regularization) and the RMSE for this predictor.

We need to solve $\min_{b_u, b_i} \sum_{(u,i)\in R}(r_{ui} - \overline{r} - b_u - b_i)^2$, where $\overline{r} = 3.25$. We first transform this cost function into the standard form $||Ab - c||^2$, as follows. Each row $A_{k.}$ of A corresponds to one (u, i) pair, and has two ones, one at position u, the other at position $u + i$. The vector b is $[b_u|b_i]^T$, and c is $[r_k - \overline{r}]^T$.

The matrix $A^T A$ is rank deficient with rank $r = 8$, so the minimum is not unique and the usual formula $(A^T A)^{-1} A^T c$ cannot be used. Instead, we can obtain the pseudoinverse $A^+ = V\Sigma^+ U^T$ through a SVD $A = U\Sigma V^T$, where Σ^+ is obtained by inverting every nonzero element of Σ. This provides *one* solution to the problem as

$$b = V\Sigma^+ U^T c. \tag{1}$$

To find this pseudoinverse we can directly use *pinv* command in octave or matlab.

We find $b_u = [-1.24, 0.52, -0.06, 0.85, -0.15]$ and $b_i = [0.23, -1.50, 1.23, -0.04]$. As this solution is not unique, if you obtain a different (b_u, b_i), verify that your root mean squared error is not more than 0.71244.

(c) Predict the missing values in the rating matrix R for the predictors in parts (a) and (b).

We first predict r_{ui} for user one and item four.

- Predictor (a): $\tilde{r}_{u_1 i_4} = b_{u_1} + \overline{r} = -1.25 + 3.25 = 2.00$
- Predictor (b): $\tilde{r}_{u_1 i_4} = b_{u_1} + b_{i_4} + \overline{r} = -1.24 - 0.04 + 3.25 = 1.97$

The other missing values are predicted similarly.

Exercise 2

You are training a machine learning model on some training data, and then evaluate the error on some separate validation data that you kept aside. You notice that the validation error is considerably larger than

the training error. Is this normal, or is there a problem? If there's a problem, how would you fix it by adjusting the regularizer weight λ? How will the two errors evolve if you change λ?

If the validation error is larger than the training error, this means that the model overfits: the error is small for data that the model has been trained on, but it does a poor job on predicting unseen data.

This implies that λ is too small, i.e., we are not penalizing the model complexity $g(\theta)$ enough.

To fix this, we need to increase λ. This will increase the training error, but decrease the validation error.

Exercise 3

(a) The most basic version of stochastic gradient descent (SGD) works as follows. There are n data points $(x_1, \ldots, x_n)$. We want to minimize a function $f(\theta; x_1, \ldots, x_n) = \sum_{i=1}^{n} f_i(\theta; x_i)$ with respect to θ (which represents the model parameters to be optimized).

Instead of performing gradient descent using the full gradient $\nabla_\theta f$, we select, for each step in the iteration, a data index $I \sim unif(1, n)$ randomly, and we update the current estimate of θ with the gradient $\nabla_\theta f_I$.

What is the expected gradient $E[\nabla_\theta f_I]$?

The expected gradient in SGD is equal to the (scaled) full gradient:

$E[\nabla_\theta f_I] = \sum_{i=1}^{n} P(I = i) \nabla_\theta f_i = n^{-1} \nabla_\theta f$.

(b) As we have seen in class, training the model on data is to solve an instance of the optimization problem above, with $x_i = (u, i, r_{ui})$, $\theta = (P, Q)$, and $f(\theta) = E(P, Q)$, i.e.,

$$(P^*, Q^*) = \arg\min_{P,Q} E(P, Q) = \arg\min_{P,Q} \sum_{(u,i)\in R} (r_{ui} - p_u^T q_i)^2 + \lambda(\|P\|^2 + \|Q\|^2).$$

Compute the full gradients $\nabla_P E(P, Q), \nabla_Q E(P, Q)$ used in gradient descent for this model.

Let us look at the element (k, v) of the gradient $\nabla_P E(P, Q)$ ($\nabla_Q E(P, Q)$ can be handled analogously). This is the derivative of the cost function $E(P, Q)$ w.r.t. $p_v(k)$, i.e., the k-th element of p_v. Specifically,

$$(\nabla_P E(P, Q))_{kv} = \frac{\partial E(P, Q)}{\partial p_v(k)} = -2 \sum_{(v,i)\in R} (r_{vi} - p_v^T q_i) q_i(k) + 2\lambda p_v(k). \tag{2}$$