# Alternating Squares

Sunday, October 28, 2018 2:55 PM

2017

## 1. [Alternating Least Squares]

In matrix factorization, we choose vectors $U_a \in \mathbb{R}^k$ for each customer $a$, and vectors $V_i \in \mathbb{R}^k$ for each product $i$, such that $U_a^\top V_i$ is a good prediction for the rating $Y_{ai}$. To optimize for these vectors, we use the alternating least-squares algorithm which takes the following form:

1. Initialize vectors $V_i \in \mathbb{R}^k$ randomly.
2. Repeat until convergence:
   a. While fixing the $V_i$, for each customer $a$, find the optimal $U_a$ minimizing
   $$\sum_{(a,i)\text{ observed}} \frac{1}{2}(Y_{ai} - U_a^\top V_i)^2 + \frac{\lambda}{2}\|U_a\|^2$$
   b. While fixing the $U_a$, for each product $i$, find the optimal $V_i$ minimizing
   $$\sum_{(a,i)\text{ observed}} \frac{1}{2}(Y_{ai} - U_a^\top V_i)^2 + \frac{\lambda}{2}\|V_i\|^2$$

For step (2a) of the algorithm, for a given customer $a$, let $Z$ be the vector of all product ratings observed from customer $a$. Let $X$ be the matrix whose $j$-th row is $V_i$ if the entry $Z_j$ is a rating for product $i$.

**a.** What is the exact solution for step (2a) of the algorithm?
Ans:
$$(X^\top X + \lambda I)^{-1} X^\top Z$$

**b.** What is the reason for using a **non-zero value for $\lambda$**?
Ans: **There are infinitely many solutions for the original training loss of the matrix factorization problem, so we use $\lambda > 0$ to ensure that we get a unique solution.**

c. Improving alternating least squares algo

To improve the prediction accuracies, we now introduce additional parameters and approximate

$$Y_{ai} \approx U_a^\top V_i + \beta_a + \gamma_i + \mu$$

where $\beta_a, \gamma_i$ are parameters that represent the bias in the ratings from customer $a$ and from product $i$ respectively, and $\mu$ is the average of all the observed ratings. We will pre-compute $\mu$ from the data, but the parameters $\beta_a, \gamma_i$ will be learned by optimization. The training loss of this new model is

$$\sum_{(a,i)\text{ observed}} \frac{1}{2}(Y_{ai} - U_a^\top V_i - \beta_a - \gamma_i - \mu)^2 + \frac{\lambda}{2}\left(\sum_a \|U_a\|^2 + \sum_i \|V_i\|^2 + \sum_a \beta_a^2 + \sum_i \gamma_i^2\right).$$

By applying coordinate descent to this problem, we get the following algorithm.

1. Initialize $V_i, \beta_a, \gamma_i$ randomly.
2. Repeat until convergence:
   a. While fixing the $V_i, \beta_a, \gamma_i$, find the optimal $U_a$.
   b. While fixing the $U_a, \beta_a, \gamma_i$, find the optimal $V_i$.
   c. While fixing the $U_a, V_i$, find the optimal $\beta_a, \gamma_i$.

In step (2c) of the algorithm, what is the exact solution for $\beta_a$?

Ans:

$$\frac{1}{1+\lambda}\sum_{(a,i)\text{ observed}}(Y_{ai} - U_a^\top V_i - \gamma_i - \mu)$$