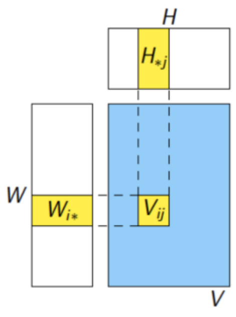
Matrix Factorization

Introduction of matrix factorization:

The collaborative matrix can be seen as the product of two small matrices on the right

$$\begin{bmatrix} 5 & 2 & 4 & 4 & 3 \\ 3 & 1 & 2 & 4 & 1 \\ 2 & & 3 & 1 & 4 \\ 2 & 5 & 4 & 3 & 5 \\ 4 & 4 & 5 & 4 & \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \\ u_{41} & u_{42} \\ u_{51} & u_{52} \end{bmatrix} \times \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} \end{bmatrix}$$



collaborative matrix Matrix w Matrix h

By this idea, we can construct two matrices w and h to approximate the actual rating matrix. Matrix w has dimension of (numberOfUser,innerDimension) and h has dimension of (innerDimension, numberOfMovies) since the collaborative matrix c has dimension of (numberOfUser, numberOfMovies).

The expected rating $\hat{r}$ of user u, movie i would be $\hat{r} = {w_u}^T h_i$

Our objective is to minimize the following result:

$$\underset{\mathbf{w},\mathbf{h}}{\operatorname{argmin}} \sum_{(u,i)\in\mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2 + \lambda\Big(\sum_i ||\mathbf{w}_i||^2 + \sum_u ||\mathbf{h}_u||^2\Big)$$

where v_{ui} is the actual rating, λ is the regression factor to remediate the overfitting problem.

Implementation of matrix factorization:

We generate matrices w and h with random value ranged from 0 to 1.

Then we use stochastic gradient descent approach to let these matrices to learn from our dataset, where λ is the regression factor,γ is the learning rate.

SGD update for random (u,i):

$$e_{ui} \leftarrow v_{ui} - \mathbf{w}_u^T \mathbf{h}_i$$
$$\mathbf{w}_u \leftarrow \mathbf{w}_u + \gamma(e_{ui}\mathbf{h}_i - \lambda \mathbf{w}_u)$$
$$\mathbf{h}_i \leftarrow \mathbf{h}_i + \gamma(e_{ui}\mathbf{w}_u - \lambda \mathbf{h}_i)$$

In each iteration, we shuffle the dataset in random order, then loop through the pair of (user u, movie i) to train our two matrices

However, the overall RMES ranges from 0.9 to 1. The performance is still needed to be improved.

Introducing the movie and user bias
The matrix factorization approach to collaborative filtering enables the flexibility in coping with various data aspects and other application-specific requirements. ${w_u}^T h_i$ tries to capture the interactions between users and movies that produce the different rating values. However, much of the observed variation in rating values is attributed to effects associated with either users or movies, known as biases or intercepts, independent of any interactions. Hence the user bias, b_u and the movie bias, b_i are introduced, the array of user bias and the array of movie bias are created with initial value of zero.

Therefore, then expected the rating would be:

$$\widehat{r_{ui}}(t+1) = \mu + b_i(t) + b_u(t) + {w_u}^T h_i(t)$$

where μ is mean value of all ratings.

the new stochastic gradient descent turns out to be:

where t is the iteration round

$$e_{ui}(t+1) \leftarrow v_{ui} - \widehat{r_{ui}}(t+1)$$

$$w_u(t+1) \leftarrow w_u(t) + \gamma \cdot ((e_{ui}(t+1) \cdot {h_i}^T(t) - \lambda \cdot w_u(t))$$

$$h_i(t+1) \leftarrow h_i(t) + \gamma \cdot \left(e_{ui}(t+1)\right) \cdot {w_u}^T(t) - \lambda \cdot h_i(t)$$

$$b_u(t+1) \leftarrow b_u(t) + \gamma \cdot (e_{ui}(t+1) - \lambda \cdot b_u(t))$$

$$b_i(t+1) \leftarrow b_i(t) + \gamma \cdot (e_{ui}(t+1) - \lambda \cdot b_i(t))$$

After using the user and movie the the overall RMES is reduced to around 0.88, which is a huge improvement.