Description of Soybean MM 4

Lozy

May 2, 2012

My solution of this match is based on the Matrix Factorization(or MF), a very popular method in Recommendation System.

However, the traditional MF is to predict one user rating of one item. My solution of this topic is to compare the ranking of one user rating of ont item. [1]

The model of this prediction of the rank is:

$$\hat{P}_{u,i,j} = \sigma(\hat{r}_{u,i} - \hat{r}_{u,j})$$

In this model, $\hat{P}_{u,i,j}$ means the probability of that item i is ranking higher than item j. And the $\hat{r}_{u,i} = b_i + (p_u + p_{u1} + p_{u2})^T q_i$, which p_u, p_{u1}, p_{u2}, q_i is the Latent factors whose dimension is set to be 20. b_i is the bias of pedigree_id. p_{u1} is about Loccd, p_{u2} is about year, p_u is about loccd * year, and q_i is about pedigree. $\sigma(x) = (1+exp(-x))^{-1}$, it is just for restricting the gap between item i and j, to be a format of probability.

Now, the goal of this topic is just:

$$\min_{B,P,Q} \sum_{(u,i,j)\in\kappa} (\hat{P}_{u,i,j} - P_{u,i,j})^2 + \frac{1}{2}\lambda(\|P\|^2 + \|Q\|^2)$$

I define $P_{u,i,j} = \sigma(r_{u,i} - r_{u,j})$.[2] I use the Stochastic Gradient Descent to train this model, every time I take a pair$\langle i, j\rangle$ from the same user in the training data and train it. Besides, the number of the pair$\langle i, j\rangle$ is too large, so I use a useful method named Negative Sampling, and I sample 4 times pairs as the training set. Oh, there is a small trick in my training, I just train the pair whose gap is larger than 5 to make the ranking more accurate.

Then I train this model 70 times, the initial learning rate($garm$) is 0.03, after 20 times training, $garm$ is set to be 0.01, and after 60 times training, it is set to be $garm/2$ every time. Before every time of training, I shuffle the list of the pair to uniform all data for the same user which avoid the overfitting all the time. Also to avoid overfitting, the regularization coefficient($lambda$) is set to be 0.01.

[1] In this problem, user is defined $Loccd * year$, and item is defined $pedigree$.

[2] I have tried some other definitions, but this is the most efficient.