Matrix Factorization

Yujie Wang

8/1/2019

Factorization Method

Gradient Descent

Nonprobabilistic

f: dimension of latent factors

$q_i \in \mathbb{R}^f$: factors associated with item i, measure the extent to which the item possesses those factors.

$p_u \in \mathbb{R}^f$: factors associated with user u, measure the extent of interest the user has in items that are high on the corresponding factors.

Matrix Factorization:

$$\hat{r}_{ui} = q_i^T p_u$$

object function 1:

$$min_{q^* p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2)$$

Probabilistic

Assumptions:

1. Conditional distribution over the observed ratings: $r_{iu}|q_i, p_u, \sigma^2 \sim N(\hat{r_{ui}}|q_i, p_u, \sigma^2)$
2. $q_i \sim N(0, \sigma_q^2)$
3. $p_u \sim N(0, \sigma_p^2)$

Using Bayes Rule

$p(q, p|r) = \frac{p(r,q,p)}{p(r)} \propto p(r, q, p) = p(r|q, p)p(q)p(p)$

Maximize $logp(q, p|r)$ is equivalent to minimizing the sum-of-squared-errors objective function with quadratic regularization terms:

object function 2:

$$E = \frac{1}{2} \sum_{i=1}^{M} \sum_{u=1}^{U} I_{iu}(r_{ui} - q_i^T p_u)^2 + \frac{\sigma}{2\sigma_q} \sum_{i=1}^{M} ||q_i||^2 + \frac{\sigma}{2\sigma_p} \sum_{u=1}^{U} ||p_u||^2$$

Alternating Least Squares

object function 3:

$$min_{q^*p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(\sum_i n_{q_i} ||q_i||^2 + \sum_u n_{p_u} ||p_u||^2)$$

- Step 1 Initialize matrix q by assigning the average rating for that movie as the first row, and small random numbers for the remaining entries.
- Step 2 Fix q, solve p by minimizing the objective function;
- Step 3 Fix p, solve q by minimizing the objective function similarly;
- Step 4 Repeat Steps 2 and 3 until a stopping criterion is satisfied.

Difficulty Summary

Number	Factorization Method	Difficulty	Paper
1	SGD to minimize object function 1	3	1
2	GD to minimize object function 2	5	3
3	Alternating Least Squares to minimize object function 3	3	4

Regularization Terms

1. penalize magnitudes

$$\sum_{(u,i)\in K} \lambda(||q_i||^2 + ||p_u||^2)$$

2. bias and intercepts: some users to give higher ratings than others, and for some items to receive higher ratings than others.

$$b_u i = \mu + b_i + b_u$$

$$\hat{r}_{ui} = b_{ui} + q_i^T p_u$$

3. treat the item bias bi and user bias as a function of time

$$\hat{r}_{ui} = \mu + b_i(t) + b_u(t) + q_i^T p_u(t)$$

Difficulty Summary

Number	Regularization Terms	Difficulty	Paper
1	penalize magnitudes	1	1
2	bias and intercepts	2	1
3	temporal dynamics	3	5

Postprocessing

1. global bias correction: Given a prediction P , if the mean of P is not equal to the mean of the test dataset, we can shift all predicted values by a fixed constant $\tau = mean(test) - mean(P)$.
2. KNN: define similarity between movie i_1 and i_2 as cosine similarity between q_{i_1} and q_{i_2}

$$s(q_{i_1}, q_{i_2}) = \frac{q_{i_1}^T q_{i_2}}{||q_{i_1}||||q_{i_2}||}$$

3. Kernel Ridge Regression:

Discard all weights p_{uk}

Define y as vector of ratings by users u;

X: for each row of X, normalized vector of factors for movie rated by user u

$$X_{.i} = \frac{q_i}{||q_i||}$$

$$y = X\beta$$

Solve ridge regession:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$$

Prediction:

$$\hat{r_i} = K(x_i^T, X)(K(X, X) + \lambda I)^{-1} y$$

4. find x^* that minimizes RMSE(P_x) where $P_x = (1 - x)P_0 + xP_1$

P_0 and P_0 are two different predictors

Difficulty Summary

Number	Postprocessing	Difficulty	Paper
1	global bias correction	1	4
2	Postprocessing SVD with KNN	2	2
3	Postprocessing SVD with kernel ridge regression	3	2
4	linearly combine predictors	1	4

Reference

1. Y. Koren. “Matrix Factorization Techniques for Recommender Systems.” Journal Computer, 42 , no. 8, 2009, pp. 30–37.
2. A. Paterek, “Improving Regularized Singular Value Decomposition for Collaborative Filtering,” Proc. KDD Cup and Workshop, ACM Press, 2007, pp. 39-42.
3. R. Salakhutdinov and A. Mnih, “Probabilistic Matrix Factorization,” Proc. Advances in Neural Information Processing Systems 20 (NIPS 07), ACM Press, 2008, pp. 1257-1264.
4. Y. Zhou et al., “Large-Scale Parallel Collaborative Filtering for the Netflix Prize,” Proc. 4th Int’l Conf. Algorithmic Aspects in Information and Management, LNCS 5034, Springer, 2008, pp. 337-348.
5. Y. Koren, “Collaborative Filtering with Temporal Dynam- ics,” Proc. 15th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD 09), ACM Press, 2009, pp. 447-455.