Introduction to Matrix Factorization Methods collaborative Filtering

User ratings Prediction

Alex Lin
Senior Architect
Intelligent Mining

Outline

- Factor analysis
- Matrix decomposition
- Matrix Factorization Model
- Minimizing Cost Function
- Common Implementation

Factor Analysis

- A procedure can help identify the factors that might be used to explain the interrelationships among the variables
- Model based approach

Refresher: Matrix Decomposition

r
5 x 6 matrix

X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}
X_{21}	X_{22}	X_{12}	X_{24}	X_{25}	X_{26}
X_{31}	X_{32}	X_{33}	X_{34}	X_{35}	X_{36}
X_{41}	X_{42}	X_{43}	X_{44}	X_{45}	X_{46}
X_{51}	X_{52}	X_{53}	X_{54}	X_{55}	X_{56}

=

q
5 x 3 matrix

a	b	c

p
3 x 6 matrix

	x				
	y				
	z				

$$X_{32} = (a, b, c) \cdot (x, y, z) = a * x + b * y + c * z$$

Rating Prediction

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

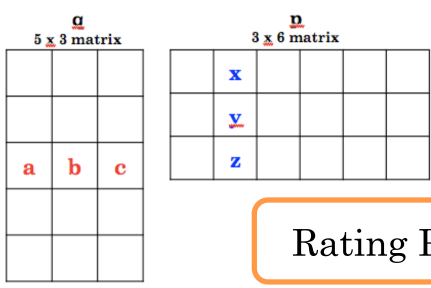
User Preference Factor Vector

Movie Preference Factor Vector

Making Prediction as Filling Missing Value

users

items	1	2	3	4	5	6	7	8	9	10	…	n
1	5		4			4				3		
2							3		?			
3	3	?		5	?			2				
4		3			3			4	3			
⋮			4				4					
m												



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

Rating Prediction

User Preference Factor Vector

Movie Preference Factor Vector

Learn Factor Vectors

users

items	1	2	3	4	5	6	7	8	9	10	…	n
1	5		4			4				3		
2							3		?			
3	3	?		5	?			2				
4		3			3			4	3			
⋮			4				4					

$$\begin{cases} 4 = U_{3\text{-}1} * I_{1\text{-}1} + U_{3\text{-}2} * I_{1\text{-}2} + U_{3\text{-}3} * I_{1\text{-}3} + U_{3\text{-}4} * I_{1\text{-}4} \\ 3 = U_{7\text{-}1} * I_{2\text{-}1} + U_{7\text{-}2} * I_{2\text{-}2} + U_{7\text{-}3} * I_{2\text{-}3} + U_{7\text{-}4} * I_{2\text{-}4} \\ \ldots\ldots \\ 3 = U_{86\text{-}1} * I_{12\text{-}1} + U_{86\text{-}2} * I_{12\text{-}2} + U_{86\text{-}3} * I_{12\text{-}3} + U_{86\text{-}4} * I_{12\text{-}4} \end{cases}$$

Note: only train on known entries

$$\begin{cases} 2X + 3Y = 5 \\ 4X - 2Y = 2 \end{cases} \qquad \begin{cases} 2X + 3Y = 5 \\ 4X - 2Y = 2 \\ 3X - 2Y = 2 \end{cases}$$

Why not use standard SVD?

- Standard SVD assumes all missing entries are zero. This leads to bad prediction accuracy, especially when dataset is extremely sparse. (98% - 99.9%)
- See Appendix for SVD
- In some published literatures, they call Matrix Factorization as SVD, but note it's NOT the same kind of classical low-rank SVD produced by svdlibc.

How to Learn Factor Vectors

- How do we learn preference factor vectors **(a, b, c)** and **(x, y, z)**?
- Minimize errors on the known ratings

$$\min_{q^*.p^*} \sum_{(u,i)\in k} (r_{ui} - x_{ui})^2$$

To learn the factor vectors (p_u and q_i)

Minimizing Cost Function (Least Squares Problem)

r_{ui} : actual rating for user u on item I
x_{ui} : predicted rating for user u on item I

proprietary material

Data Normalization

- Remove Global mean

users

items	1	2	3	4	5	6	7	8	9	10	…	n
1	1.5		-.9			-.2				.49		
2							.79		?			
3	0.6	?		.46	?			-.4				
4		.39			.82			.76	.69			
⋮			.52				.8					
m												

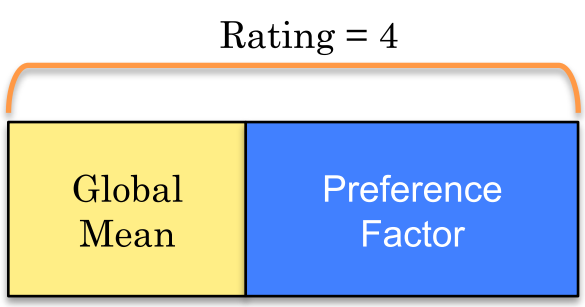
proprietary material

Factorization Model

- Only Preference factors

$$\min_{q^*.p^*} \sum_{(u,i)\in k} (r_{ui} - \mu - q_i^T p_u)^2$$

To learn the factor vectors (p_u and q_i)



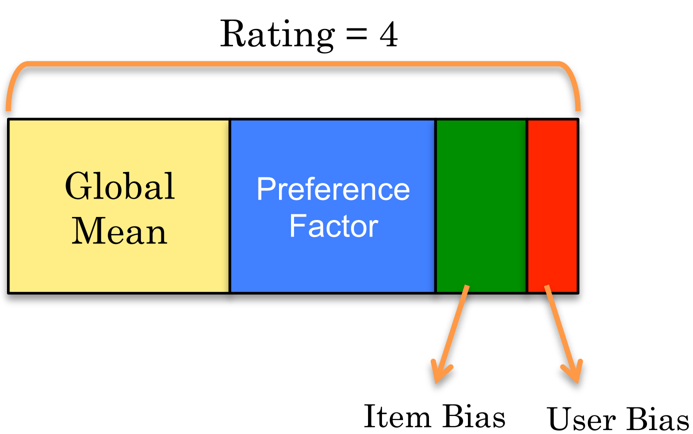
r_{ui} : actual rating of user u on item I
u : training rating average
b_u : user u user bias
b_i : item i item bias
q_i : latent factor array of item i
p_u : later factor array of user u

Adding Item Bias and User Bias

- Add Item bias and User bias as parameters

$$\min_{q^*.p^*} \sum_{(u,i)\in k} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2$$

To learn Item bias and User bias



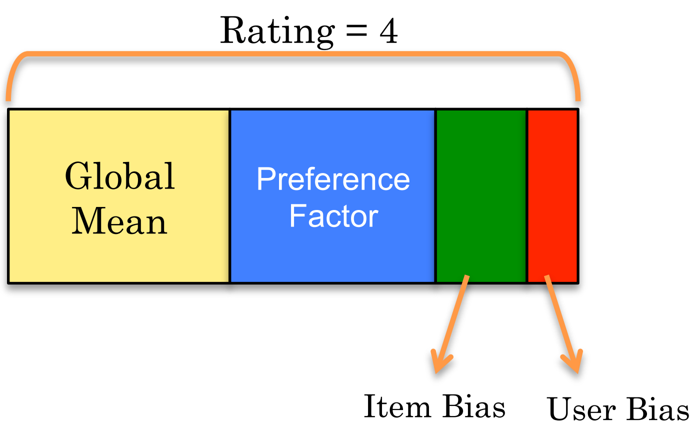
r_{ui} : actual rating of user u on item I
u : training rating average
b_u : user u user bias
b_i : item i item bias
q_i : latent factor array of item i
p_u : later factor array of user u

proprietary material

Regularization

- To prevent model overfitting

$$\min_{q^*,p^*} \sum_{(u,i)\in k} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \underbrace{\lambda(\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2)}_{\text{Regularization to prevent overfitting}}$$



r_{ui} : actual rating of user u on item I
u : training rating average
b_u : user u user bias
b_i : item i item bias
q_i : latent factor array of item i
p_u : later factor array of user u
λ : regularization Parameters

Optimize Factor Vectors

- Find optimal factor vectors - minimizing cost function
- Algorithms:
  - Stochastic gradient descent
  - Others: Alternating least squares etc..
- Most frequently use:
  - Stochastic gradient descent

Matrix Factorization Tuning

- Number of Factors in the Preference vectors
- Learning Rate of Gradient Descent
  - Best result usually coming from different learning rate for different parameter. Especially user/item bias terms.
- Parameters in Factorization Model
  - Time dependent parameters
  - Seasonality dependent parameters
- Many other considerations !

High-Level Implementation Steps

- Construct User-Item Matrix (sparse data structure!)
- Define factorization model - Cost function
- Take out global mean
- Decide what parameters in the model. (bias, preference factor, anything else? SVD++)
- Minimizing cost function - model fitting
  - Stochastic gradient descent
  - Alternating least squares
- Assemble the predictions
- Evaluate predictions (RMSE, MAE etc..)
- Continue to tune the model

Thank you

- Any question or comment?

Appendix

- Stochastic Gradient Descent
- Batch Gradient Descent
- Singular Value Decomposition (SVD)

Stochastic Gradient Descent

Repeat Until Convergence {

for i=1 to m in random order {

$$\theta_j := \theta_j + \alpha\underbrace{(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}}_{\text{partial derivative term}} \quad \text{(for every j)}$$

}

}

Your code Here:

proprietary material

Batch Gradient Descent

Repeat Until Convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \underbrace{(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}}_{\text{partial derivative term}} \quad \text{(for every j)}$$

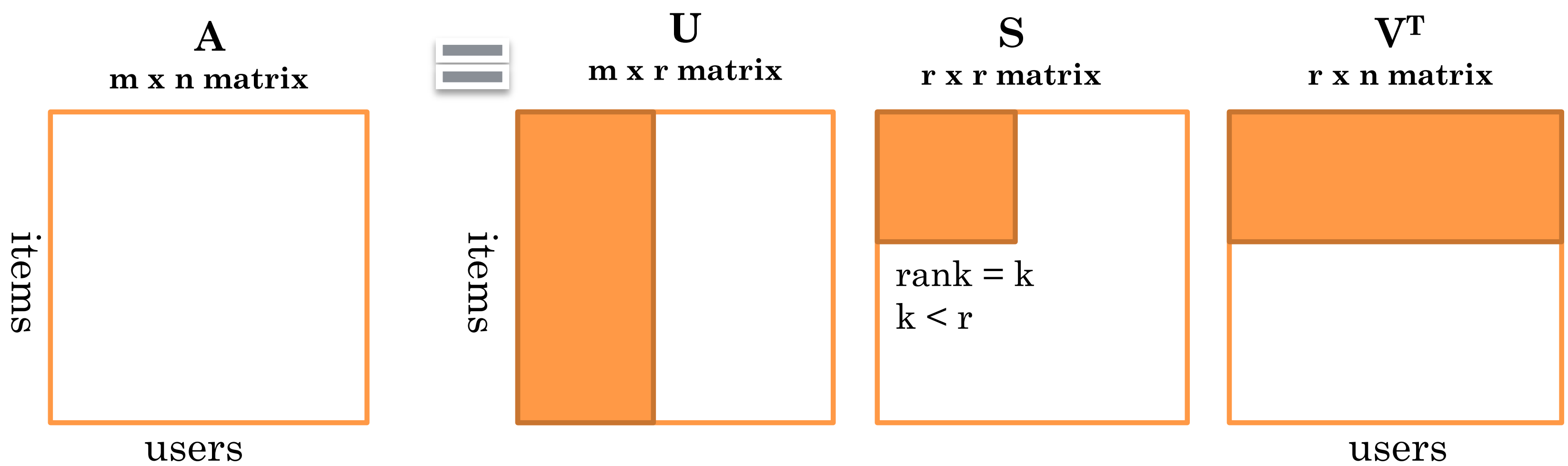
}

Your code Here:

proprietary material

Singular Value Decomposition (SVD)

$$A = U \times S \times V^T$$



$$A_k = U_k \times S_k \times V_k^T$$

proprietary material