Collaborative Filtering Algorithms Evaluation

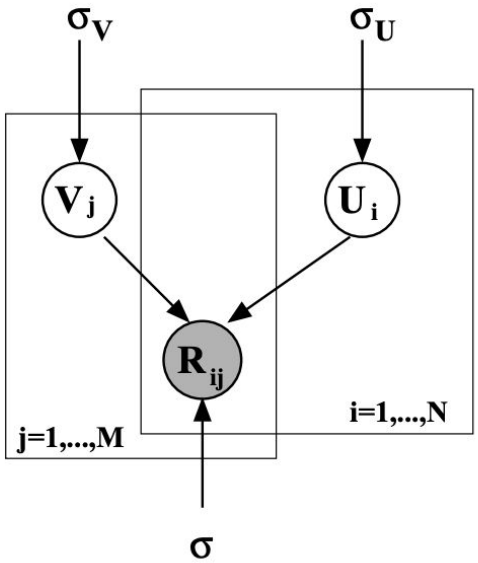
By group 8:

Shuxin Chen (sc4599)
Jia Li (jl5520)
Wenfeng Lyu (wl2733)
Daniel Schmidle (dps2150)
Yuyao Wang (yw3395)

Project Overview

- Project goal:
  - implement, evaluate, and compare algorithms A2 & A3 given P3 for collaborative filtering
- Algorithms:
  - A2: Gradient Descent with Probabilistic Assumptions (PMF)
  - A3: Alternating Least Squares (ALS)
- Model evaluation without Postprocessing
- Postprocessing:
  - SVD with Kernal Ridge Regression (KRR: P3)
- Model evaluation with Postprocessing

Algorithm A2: Probabilistic Matrix Factorization (PMF)



- Factor- based method for collaborative filtering
- Assigned a D-dimensional latent feature vector for each user and movie, denoted as Ui , Vj $\in \mathbb{R}^D$

$$U_i \sim \mathcal{N}(0, \sigma_U^2 \mathbf{I}), \qquad i = 1 \dots N,$$
$$V_j \sim \mathcal{N}(0, \sigma_V^2 \mathbf{I}), \qquad j = 1 \dots M,$$

- Model each ratings as the inner product of corresponding latent features, $i.e.\ R_{ij} \approx U_i' V_j$
- The conditional distribution is assumed:

$$p(\mathbf{R}|\mathbf{U}, \mathbf{V}, \alpha) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}\left(R_{ij}|U_i' V_j, \alpha^{-1}\right)\right]^{I_{ij}}$$

where

$$\mathbf{U} \in \mathbb{R}^{D \times N} \qquad \mathbf{V} \in \mathbb{R}^{\tilde{D} \times \tilde{M}}$$

Algorithm A2: Probabilistic Matrix Factorization (PMF)

The learning procedure of the model:

$$\mathbf{U}, \mathbf{V} = \arg\min_{\mathbf{U},\mathbf{V}} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} \left(R_{ij} - U_i' V_j\right)^2 + \frac{\lambda_U}{2} \sum_{i}^{N} \|U_i\|^2 + \frac{\lambda_V}{2} \sum_{j}^{N} \|V_j\|^2.$$

where the above optimization can be solved using gradient descent

Algorithm A3: Alternating Least Squares (ALS)

Solving the objective function:

$$L = \sum_{m,n} (R_{mn} - U_m^T \cdot V_n)^2 + \lambda \sum_m ||U_m||^2 + \lambda \sum_n ||V_n||^2$$

where $R_{mn} = U_m^T \cdot V_n$

Step as follows:

1. Initialize matrix V by assigning the average rating for that movie as the first row, and small random numbers for the remaining entries
2. Fix V, solve U by minimizing the objective function (the sum of squared errors)
3. Fix U, solve V by minimizing the objective function similarly
4. Repeat step 2&3 until a stopping criterion is satisfied

Postprocessing: SVD with Kernel Ridge Regression

- Idea: discard all weights u_{ik} after training and, for each user i, predict y_{ij} using $x_{j_2} = \frac{v_j}{||v_j||}$ as predictors
- We then predict y using ridge regression:

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

$$\hat{y}_i = x_i^{\mathrm{T}} \hat{\beta}$$

- Predictions in kernel ridge regression:

$$\hat{y}_j = K(x_j^{\mathrm{T}}, X)(K(X, X) + \lambda I)^{-1} y$$

- In place of the scalar product we use a Gaussian kernel:

$$K(x_j^{\mathrm{T}}, x_k^{\mathrm{T}}) = exp(2(x_j^{\mathrm{T}} x_k - 1))$$

Model evaluation: method

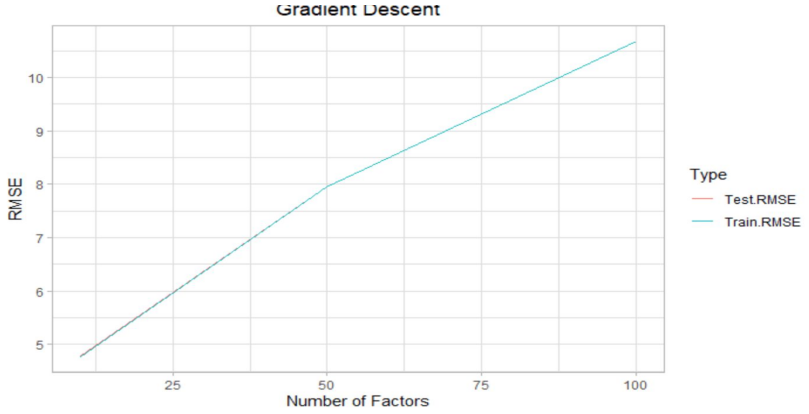
Root mean square error (RMSE) is given by:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(d_i - \hat{d}_i)^2}$$

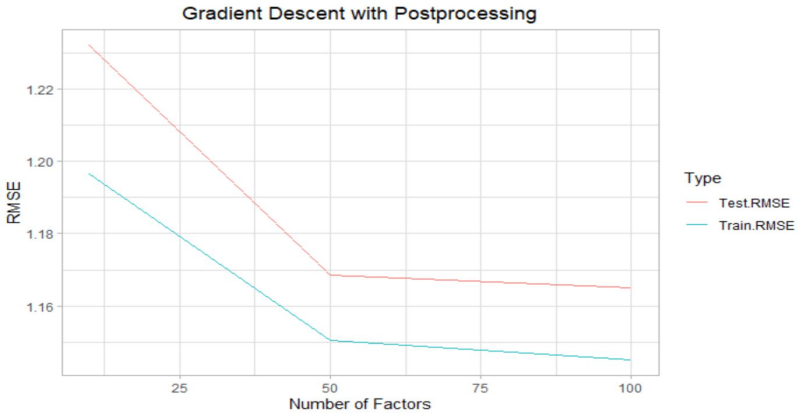
d_i is the actual rating
$\hat{d}_i$ is the predicted rating
n is the amount of ratings

- Computes the mean value of all the differences squared between the true and the predicted ratings and then proceeds to calculate the square root of the result
- RMSE metric most valuable when significantly large errors are unwanted

Model evaluation: PMF results (A2)

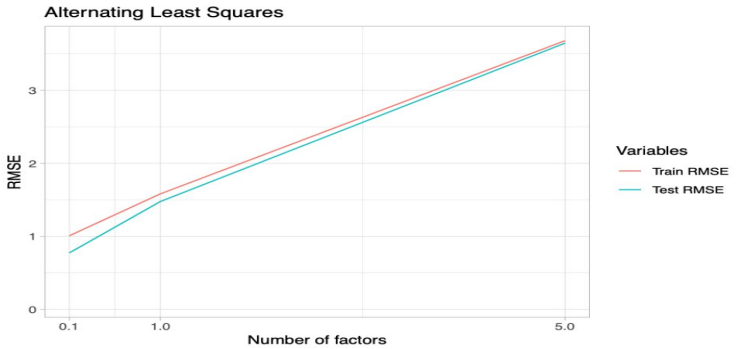


factor	Type	RMSE
10	Train.RMSE	4.771120134
10	Test.RMSE	4.789309655
50	Train.RMSE	7.942648782
50	Test.RMSE	7.942138333
100	Train.RMSE	10.66689553
100	Test.RMSE	10.6712375

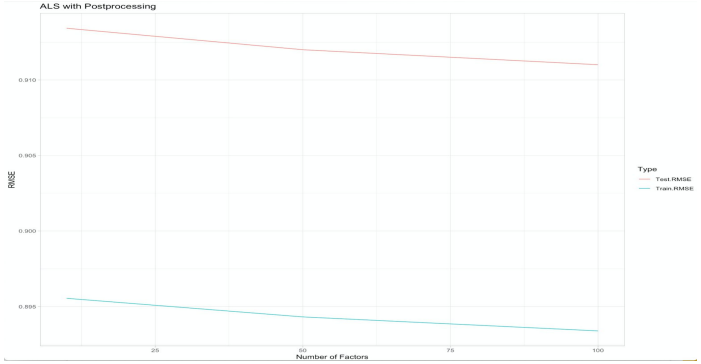


	X	factor	Type	RMSE
1	1	10	Train.RMSE	1.196468
2	2	10	Test.RMSE	1.232034
3	3	50	Train.RMSE	1.150450461
4	4	50	Test.RMSE	1.168556656
5	5	100	Train.RMSE	1.145147925
6	6	100	Test.RMSE	1.165039263

Model evaluation: ALS results (A3)




```
##   Factors Lambda      RMSE Variable
## 1     100    0.1 0.7737497 training
## 2     100    1.0 1.4774650 training
## 3     100    5.0 3.6467150 training
## 4     100    0.1 1.0083640     test
## 5     100    1.0 1.5823750     test
## 6     100    5.0 3.6801430     test
```



	factor	Type	RMSE
1	10	Train.RMSE	0.8955452
2	10	Test.RMSE	0.9134247
3	50	Train.RMSE	0.8943138
4	50	Test.RMSE	0.9120027
5	100	Train.RMSE	0.8933866
6	100	Test.RMSE	0.9110096

Evaluation Summary

	Gradient Descent with Probabilistic Assumptions (A2)	Alternating Least Squares (A3)
RMSE without Postprocessing	Train error: 4.771120134 Test error: 4.789309655	Train error: 0.7737497 Test error: 1.0083640
RMSE with Postprocessing (P3)	Train error: 1.196468 Test error: 1.232034	Train error: 0.8933866 Test error: 0.9110096
Model Training Time	0.92 secs/13.32 mins	0.81 secs/14.6146 mins

Conclusion

- Gradient Descent with Probabilistic Assumptions (A2) vs ALS (A3)
  - The ALS model outperforms gradient descent with probabilistic assumptions in general
- With/without SVD with kernel ridge regression (P3)
  - The test/train errors for both models A2&A3 have decreased using SVD with kernel ridge regression
  - The Alternating Least Squares (A3) model given P3 has higher prediction accuracy than Gradient Descent with Probabilistic Assumptions given P3

Reference

- Xiong, L., Chen, X., Huang, T.-K., Schneider, J., & Carbonell, J. G. (n.d.). Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. Retrieved April 2020, from https://www.cs.cmu.edu/~jgc/publication/PublicationPDF/Temporal_Collaborative_Filtering_With_Bayesian_Probabilidtic_Tensor_Factorization.pdf
- Zhong, Y., Wilkinson, D., Schreiber, R., & Pan, R. (n.d.). Large-scale Parallel Collaborative Filtering for the Netflix Prize. Retrieved April 2020
- Paterek, A. (n.d.). Improving regularized singular value decomposition for collaborative filtering. Retrieved April 2020

Thank you !