

Low-rank neighborhood model

Kabbur, KDD 2013

- Instead of learning S , two low dimensional matrices $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{n \times k}$ are learnt where $S = UV^T$

$$\hat{U}, \hat{V} = \min_{U, V} \|R - RU\mathbf{V}^T\|_F^2 + \lambda \|\mathbf{UV}^T\|_F^2 \quad s.t. \quad \text{diag}(\mathbf{UV}^T) = 0$$

- Parameters U and V are k -dimensional
 - $k \ll |\text{items}|$

Assuming $n = 1M$ and $k = 10$, then $U = 1M \times 10$ and $V = 1M \times 10$ require ~0.08Gb memory

Other general limitations ...

- Updating the similarity matrix with new observations requires updating the whole matrix
- As a general problem in recommender systems, these models also suffer from different types of bias
 - e.g., popularity bias

Summary

- **SLIM** improves **ItemKNN** in computing similarity matrix and computational complexity
- **EASE^R** improves **SLIM** in further improving computational complexity
- **Low-rank neighborhood model** improves **SLIM** and **EASE^R** with respect to the memory required to store the model parameters

Resources

- Dynamically updating regression model with new interaction data
 - *Efficient similarity computation for collaborative filtering in dynamic environments (RecSys 2019)*
 - *Embarrassingly shallow auto-encoders for dynamic collaborative filtering (UMUAI 2022)*
- Online learning of recommendation system
 - *Cascading bandits for large-scale recommendation problems (UAI 2016)*
 - *Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank (UAI 2020)*

Next session ...

- Evaluation of recommender systems
 - Evaluation paradigms
 - Metrics for rating prediction task
 - Metrics for ranking tasks