The Fifteenth International Conference on
Internet and Web Applications and Services

Paper 20017:

Improving DMF with Hybrid Loss Function and Applying CF-NADE to The MOOC Recommendation System

Ngoc-Thanh Le
lnthanh@fit.hcmus.edu.com

Ngoc-Khai Nguyen
1612909@student.hcmus.edu.com

Nhat-Vinh Vo
1612815@student.hcmus.edu.vn

Hoai-Bac Le
lhbac@fit.hcmus.edu.com

Faculty of Information Technology
University of Science, VNU-HCM
Ho Chi Minh City, Vietnam

September 27, 2020 to October 01, 2020 - Lisbon, Portugal

CONTENTS

1. Motivation
2. Deep matrix factorization model (DMF)
3. Hybrid Deep matrix factorization model (Hybrid-DMF)
4. Neural Autoregressive Distribution Estimator for Collaborative Filtering (CF-NADE)
5. Experiment
6. Conclusion and Future works

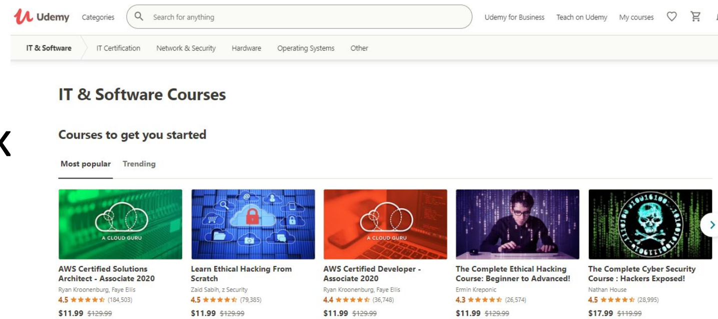
CONTENTS

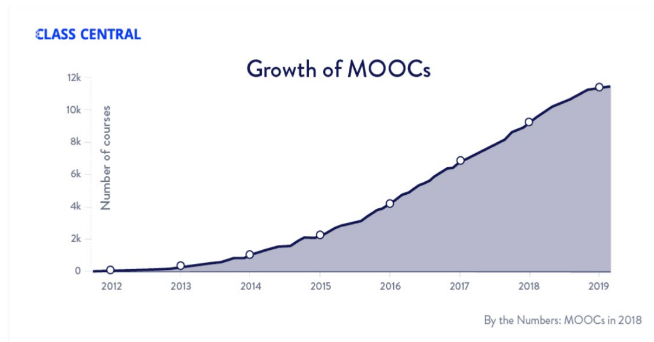
1. **Motivation**
2. Deep matrix factorization model (DMF)
3. Hybrid Deep matrix factorization model (Hybrid-DMF)
4. Neural Autoregressive Distribution Estimator for Collaborative Filtering (CF-NADE)
5. Experiment
6. Conclusion and Future works

Motivation

- There are many MOOC platforms like Coursera, Edx Udemy, etc.,with millions courses.
- These MOOC plaform attracts hundred millions learners.

→ How to recommend users to choose the appropriate course.





From https://www.classcentral.com

Motivation

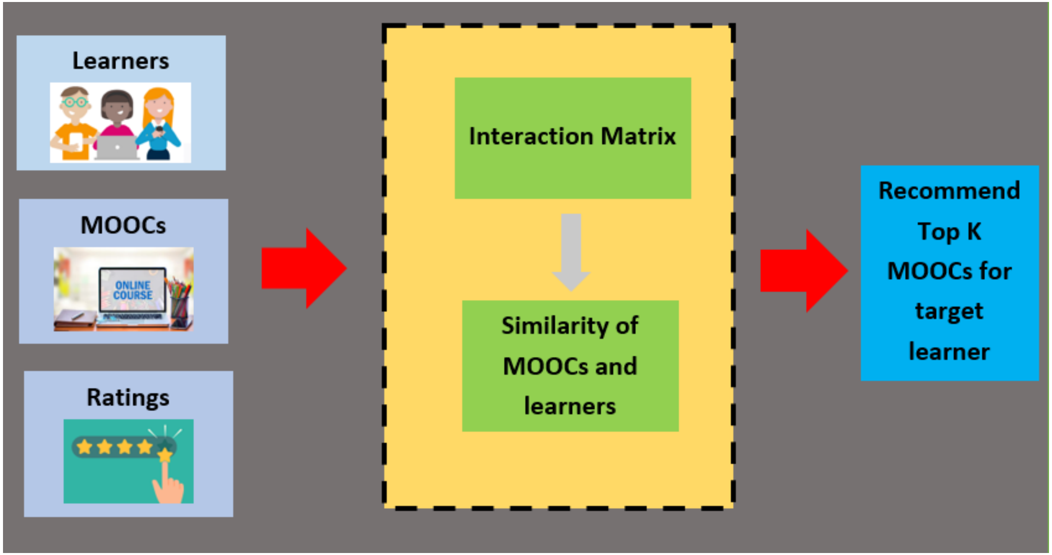
- There are few works about MOOC Recommendation system using deep learning.
- Neural Autoregressive Distribution Estimator for Collaborative Filtering(CF-NADE) and Deep Matrix Factorization (DMF) have not been used in Recommendation system.

→ Apply the CF-NADE and DMF model with the improved loss function to the MOOCs suggestion.

CONTENTS

1. Motivation
2. **Deep matrix factorization model (DMF)**
3. Hybrid Deep matrix factorization model (Hybrid-DMF)
4. Neural Autoregressive Distribution Estimator for Collaborative Filtering (CF-NADE)
5. Experiment
6. Conclusion and Future works

MOOC Recommendation with Deep Matrix Factorization

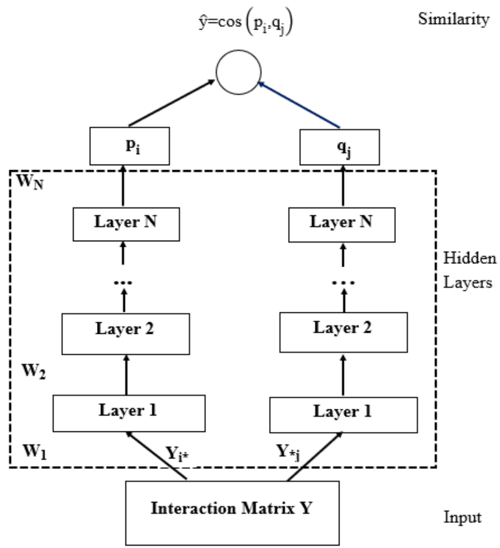


Notation

DMF is proposed by Xue et al., 2017 [1].

Assume:

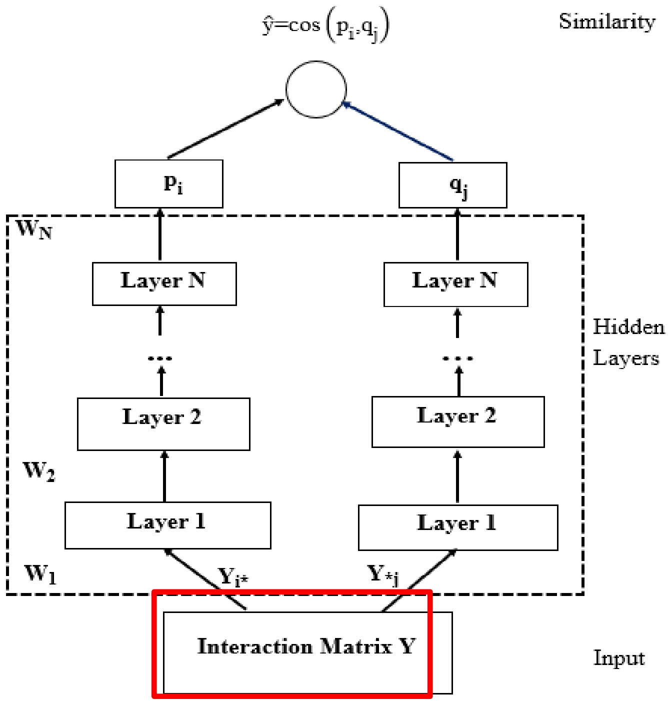
- Set U includes M users: **$U=\{u_1,u_2,\ldots,u_M\}$**.
- Set V includes N items: **$V=\{v_1,v_2,\ldots,v_N\}$**.
- **$R \in \mathbb{R}^{MxN}$** is the rating matrix with **R_{ij}** is rating of user **i** for item **j**, **unk** is unknown rating.
- **i**, **j** is the user and item in **U**, **V**



Interaction matrix

Rating interaction matrix.

$$Y_{ij}=\begin{cases} 0, \text{ if } \mathbf{R_{ij}}=\text{unk} \\ \mathbf{R_{ij}}, \text{ otherwise} \end{cases}$$



Hidden Layers

- Row **i** of matrix **Y** is $\mathbf{Y_{i*}}$, column **j** of the matrix is $\mathbf{Y_{*j}}$.

  This model has two MLPs, one for users and one for items.
- Multi-layer perceptron (MLP) uses.

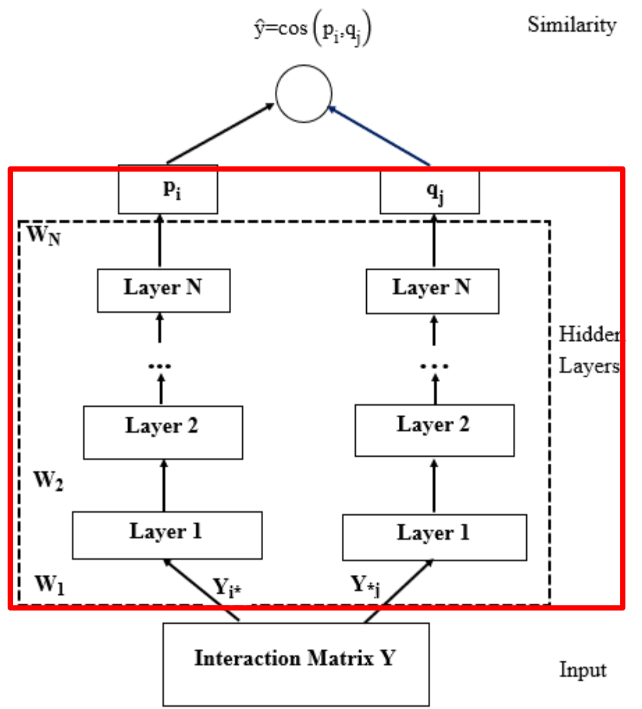
$$l_1 = W_1 x$$

$$l_i = f(W_{i-1} l_{i-1} + b_i);\ i=2,\ldots N-1$$

$$y = f(W_N l_{N-1} + b_N)$$

- The activation function is ReLU.

$$f(x) = \max(0,x)$$

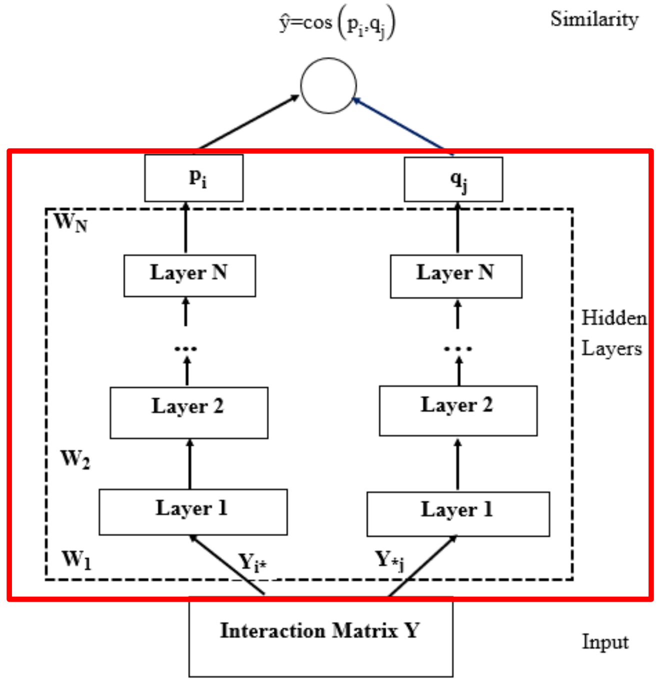


Hidden Layers

In other words, the user and item vector are mapped into low dimensional vectors in latent space using two MLPs.

$$p_i = f_{\theta_N^U}\left(\cdots f_{\theta_3^U}\left(W_{U2} f_{\theta_2^U}\left(Y_{i*} W_{U1}\right)\right)\cdots\right)$$

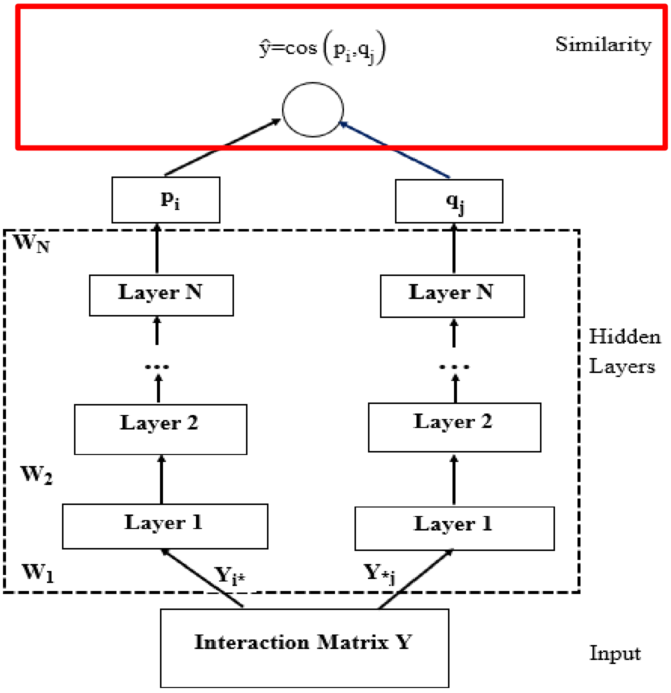
$$q_j = f_{\theta_N^I}\left(\cdots f_{\theta_3^I}\left(W_{V2} f_{\theta_2^I}\left(Y_{*j}^T W_{V1}\right)\right)\cdots\right)$$



Cosine similarity

Then, we calculate the cosine similarity of two latent representations p_i and q_j.

$$\text{cosine}\left(p_i, q_j\right) = \frac{p_i^T . q_j}{\|p_i\| . \|q_j\|}$$



CONTENTS

1. Motivation
2. Deep matrix factorization model (DMF)
3. **Hybrid Deep matrix factorization model (Hybrid-DMF)**
4. Neural Autoregressive Distribution Estimator for Collaborative Filtering (CF-NADE)
5. Experiment
6. Conclusion and Future works

Normalized cross-entropy loss function

- Normalized cross-entropy loss function (NCE) combines cross-entropy and max rating [1]:

$$L_{NCE} = -\sum_{(i,j)\in Y} \left(\frac{Y_{ij}}{\mathbf{max}(\mathbf{Rating})} \log \widehat{Y}_{ij} + \left(1 - \frac{Y_{ij}}{\mathbf{max}(\mathbf{Rating})}\right) \log\left(1-\widehat{Y}_{ij}\right) \right)$$

- $\frac{Y_{ij}}{\max(\text{Rating})} \in [0;1]$, so it is called Normalized cross-entropy (NCE)
- In our experiment, we use max(Rating) = 5 because 5 is the max rating.

L2 loss function

- L2 loss function fits in solving the overfitting problem.

$$L2 = \frac{\sum_i^m w_i^2}{2}$$

Where:

- $w_i^2 = \sum_j^N w_{ij}^2$, and w_{ij} is the weight of the training instance (i,j);

- N is the dimension of w_{ij}

Hybrid loss function

- Hybrid loss function combines Normalized cross-entropy loss function and L2 loss function.
- Hybrid loss function:

$$L = -\underbrace{\sum_{(i,j)\in Y^{+}\cup Y^{-}} \left(\frac{Y_{ij}}{\max(\text{Rating})}\log\hat{Y}_{ij} + \left(1-\frac{Y_{ij}}{\max(\text{Rating})}\right)\log\left(1-\hat{Y}_{ij}\right)\right)}_{L_{NCE}} + \beta.\underbrace{\frac{\sum_{i}^{M}\sum_{j}^{N} w_{ij}^{2}}{2}}_{L2}$$

CONTENTS

1. Motivation
2. Deep matrix factorization model (DMF)
3. Hybrid Deep matrix factorization model (Hybrid-DMF)
4. **Neural Autoregressive Distribution Estimator for Collaborative Filtering (CF-NADE)**
5. Experiment
6. Conclusion and Future works

Notation

- CF-NADE is proposed by Zheng et al., 2016 [2].
- **Assuming:**

- There are M courses and N users, the user ratings are from 1 to K.
- Each user rated D courses and $D \ll M$. With any user u, we will have the rating vector $r^u = \left(r^u_{m_{o_1}}, r^u_{m_{o_2}}, \ldots, r^u_{m_{o_D}}\right)$, where o is the permutation of (1, 2, …, D), $r^u_{m_{o_i}} \in \{1,2,\ldots,K\}$ is present for the rating of user u and item m_{o_i}.

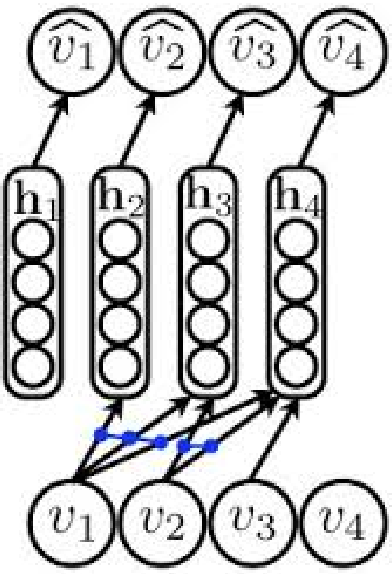
CF-NADE basic model (NADE)

- The probability of the rating vector

$$p(\mathbf{r}) = \prod_{i=1}^{D} p\left(r_{m_{o_i}} \middle| r_{m_{o_{<i}}}\right)$$

- Hidden presentation in a hidden layer

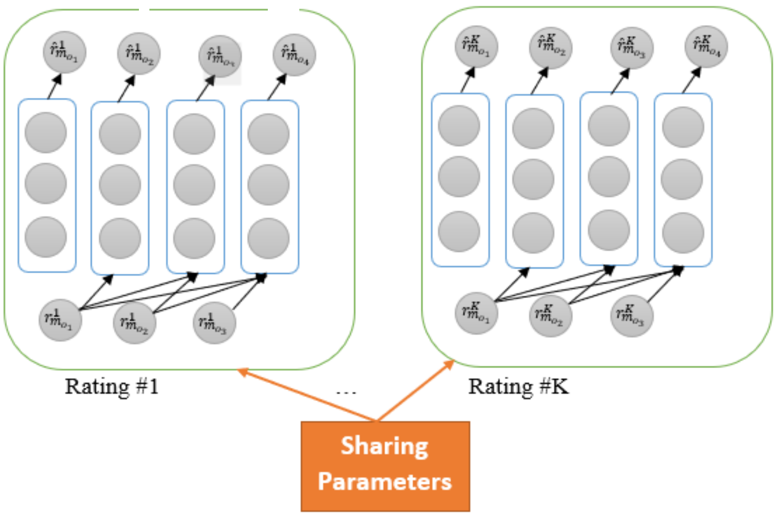
$$h\left(r_{m_{o_{<i}}}\right) = g\left(c + \sum_{j<i} \sum_{k=1}^{r_{m_{o_j}}} W^{k}_{:,m_{o_j}}\right)$$



CF-NADE model with sharing parameters

$$h(r_{mo_{<i}}) = g\left(c + \sum_{j<i} W_{:,\, mo_j}^{r_{mo_j}}\right) \longrightarrow h(r_{mo_{<i}}) = g\left(c + \sum_{j<i} \sum_{k=1}^{r_{mo_j}} W_{:,\, mo_j}^{k}\right)$$

$$s_{mo_i}^{k}(r_{mo_{<i}}) = b_{mo_i}^{k} + V_{mo_j,:}^{k} h(r_{mo_{<i}}) \longrightarrow s_{mo_i}^{k}(r_{mo_{<i}}) = \sum_{j \le k}\left(b_{mo_i}^{j} + V_{mo_i,:}^{j} h(r_{mo_{<i}})\right)$$



Ordinal Cost Function

- Assume that user rates k, then the rating from 1 to k has priority increase, and the value from k to K has priority decrease.

$$p\left(r_{m_{o_i}} = k \middle| r_{m_{o_{<i}}}\right) = \prod_{j=k}^{1} \frac{\exp\left(s_{m_{o_i}}^{j}\right)}{\sum_{t=1}^{j} \exp\left(s_{m_{o_i}}^{t}\right)} \prod_{j=k}^{K} \frac{\exp\left(s_{m_{o_i}}^{j}\right)}{\sum_{t=j}^{K} \exp\left(s_{m_{o_i}}^{t}\right)}$$

- Cost function:

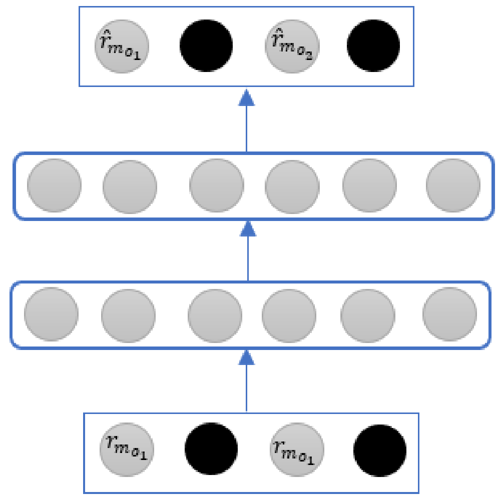
$$C_{hybrid} = (1-\lambda) C_{reg} + \lambda C_{ord}$$

Extend CF-NADE to a deep neural network

- When added a hidden layer to the model, the calculation formula of that layer:

$$h^{(l)}\left(r_{m_{o_{<i}}}\right)=g\left(c^{(l)}+W^{(l)}h^{(l-1)}\left(r_{m_{o_{<i}}}\right)\right)$$

where $l = 2, \ldots, L$ correspond to the hidden layers and the conditional probability $p\left(r_{m_{o_{<i}}}\right)$ is computed based on $h^{(L)}\left(r_{m_{o_{<i}}}\right)$.



CONTENTS

1. Motivation
2. Deep matrix factorization model (DMF)
3. Hybrid Deep matrix factorization model (Hybrid-DMF)
4. Neural Autoregressive Distribution Estimator for Collaborative Filtering (CF-NADE)
5. **Experiment**
6. Conclusion and Future works

Dataset

- Travel-well dataset [3] is used for our experiment.
- The Travel-well was collected from the LRE portal includes 20 content providers from Europe and elsewhere.
- In our experiment, we only use rating information with 75 learners.

#learners (#users)	#courses (#items)	#ratings	density
75	1608	2156	0.0178

Metric

- ***Normalized Discounted Cumulative Gain*** **(*NDCG*)** evaluate the ranking performance of the relevance courses [4].

$$NDCG@K = Z_K \sum_{k=1}^{K} \frac{2^{r_i} - 1}{\log_2(i+1)}$$

where Z_K is the ideal ranking has a value of 1; r_i is the graded relevance of item at position i.

- **Root mean square error (RMSE)**.

$$RMSE = \sqrt{\frac{\sum_{i,j}^{M,N} \left(r_{ij} - \hat{r}_{ij}\right)}{\#ratings}}$$

Parameter settings

- **Hybrid-DMF, DMF**

+ Hyperparameters: learning rates = 10^{-4}, max epoch = 30, batch size = 256, early stopping = 5, the latent factor = 64.
+ Requirements: python = 3.7.6, Tensorflow-gpu=1.5.0, numpy = 2.1.0.

- **CF-NADE**

+ Hyperparameter: learning rate = 10^{-3}, Hidden unit = 500, epochs = 20.
+ Requirements: Python 3.6.8. Dependence packages: Tensorflow (2.1.0), Tensorflow-gpu (2.1.0), Keras (2.0.8), Pyspark (2.4.1).

Results

We use five algorithms for evaluations (3 classical algorithms and 2 deep learning models):

+ Neighborhood-based collaborative filtering methods on item-based (IBCF) [5]

+ Neighborhood-based collaborative filtering methods on user-based (Pearson correlation) [5].

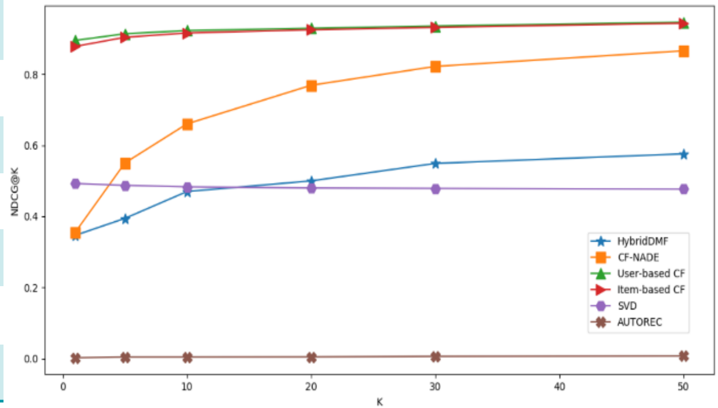
+ Single value decomposition (SVD) [6]

+ Probabilistic Matrix Factorization (PMF) [7]

+ AutoEncoder based on Collaborative Filtering (AutoRec) [8].

Results - NDCG

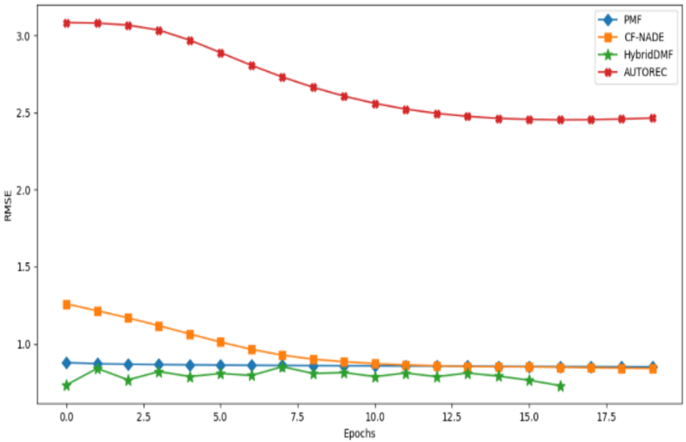
K	Hybrid-DMF	CF-NADE	Auto Rec	SVD
1	**0.3467**	**0.3554**	0.0019	0.4927
5	**0.3945**	**0.5505**	0.0039	0.4875
10	**0.4701**	**0.6606**	0.0040	0.4833
20	**0.5000**	**0.7694**	0.0043	0.4800
30	**0.5493**	**0.8225**	0.0059	0.4789
50	**0.5762**	**0.8665**	0.0070	0.4770



- Detailed results with the NDCG@K metric with K = [1, 5, 10, 20, 30, 50] of the Hybrid-DMF, CF-NADE, AutoRec, SVD, IBCF and UBCF models.

Results - RMSE

Models/Algorithms	RMSE
AutoRec	2.50037
SVD	0.9063
PMF	0.8651
CF-NADE	**0.8283**
Hybrid-DMF	**0.7916**



- Detailed results with the RMSE of the AutoRec, SVD, PMF, Hybrid-DMF, CF-NADE models.
- Hybrid-DMF and CF-NADE gives the best result.

CONTENTS

1. Motivation
2. Deep matrix factorization model (DMF)
3. Hybrid Deep matrix factorization model (Hybrid-DMF)
4. Neural Autoregressive Distribution Estimator for Collaborative Filtering (CF-NADE)
5. Experiment
6. **Conclusion and Future works**

Conclusion and Future works

- We improved the DMF model with a new loss function (Hybrid-DMF) and combined with the CF-NADE model for the MOOC recommendation system. The results show that the proposed approach is better than the other models with RMSE and NDCG@K measurements when evaluated on the travel-well data set.

- In the future, we will continue to improve DMF with some other loss function and integrate implicit feedback such as the click, tagging, side information. Improving CF-NADE can be done by implicit feedback information, such as user tagging for each course to improve the accuracy of the model.

References

1. Xue, H.J., Dai, X., Zhang, J., Huang, S. and Chen, J., 2017, August. Deep Matrix Factorization Models for Recommender Systems. In IJCAI (pp. 3203-3209).
2. Zheng, Y., Tang, B., Ding, W. and Zhou, H., 2016. "A neural autoregressive approach to collaborative filtering". arXiv preprint arXiv:1605.09477.
3. Verbert, Katrien, et al. "Dataset-driven research for improving recommender systems for learning". Proceedings of the 1st International Conference on Learning Analytics and Knowledge. 2011.
4. He, X., Chen, T., Kan, M.Y. and Chen, X., 2015, October. Trirank: Review-aware explainable recommendation by modeling aspects. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 1661-1670).

References

5. Aggarwal C.C., 2016. “Neighborhood-based collaborative filtering”. In Recommender systems (pp. 29-70). Springer, Cham.
6. Zhang, Sheng, et al. "Using singular value decomposition approximation for collaborative filtering." Seventh IEEE International Conference on E-Commerce Technology (CEC'05). IEEE, 2005.
7. Mnih, A. and Salakhutdinov, R.R., 2008. “Probabilistic matrix factorization”. In Advances in neural information processing systems (pp. 1257-1264).
8. Sedhain, S., Menon, A.K., Sanner, S. and Xie, L., 2015, May. “Autorec: Autoencoders meet collaborative filtering”. In Proceedings of the 24th international conference on World Wide Web (pp. 111-112).

Thank you