Neural Representations in Hybrid Recommender Systems: Prediction versus Regularization

Ramin Raziperchikolaei, **Tianyu Li**, and **Young-joo Chung**
Rakuten Institute of Technology, Rakuten, Inc.
{ramin.raziperchikola,tianyu.li,youngjoo.chung}@rakuten.com

Problem definition. Hybrid recommender systems use user feedback on items and user/item *side information* to help users identify the items that best fit their personal tastes [1]. Side information includes the content of items (e.g., category, title, description, etc.) and profile of users (e.g., age, location, gender, etc.). We focus on predicting explicit feedback (e.g., a ratings beetwen 1 and 5).

Assume we have a sparse rating matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, where m and n are the number of users and items, respectively. $R_{jk} > 0$ is the rating of the user j on the item k, and $R_{jk} = 0$ means the rating is unknown. Assume the side information of all the users and items are represented by $\mathbf{X}$ and $\mathbf{Y}$, respectively. The goal is to predict the unknown ratings of the matrix $\mathbf{R}$.

Autoencoder-based methods The most widely-used neural network structure in recommender systems has been (denoising) autoencoders [2, 7–12]. These methods define $\mathbf{g}^u()$, $\mathbf{f}^u()$, $\mathbf{g}^i()$, $\mathbf{f}^i()$ as the user's encoder, user's decoder, item's encoder, and item's decoder, respectively. The output of the encoders $\mathbf{g}^u()$ and $\mathbf{g}^i()$ are the learned neural representations of the users and items. They also consider $\mathbf{U} \in \mathbb{R}^{m \times d}$ and $\mathbf{V} \in \mathbb{R}^{n \times d}$ as the d-dimensional representations of the users and items, respectively. Their objective function can then be written as:

$$\min_{\mathbf{U},\mathbf{V},\boldsymbol{\theta}} Q_{\text{rec}}(\boldsymbol{\theta}) + \lambda_1 Q_{\text{MF}}(\mathbf{U},\mathbf{V}) + \lambda_2 ||\mathbf{U} - \mathbf{g}^u(\mathbf{R},\mathbf{X})||^2 + \lambda_3 ||\mathbf{V} - \mathbf{g}^i(\mathbf{R},\mathbf{Y})||^2, \tag{1}$$

where the reconstruction loss Q_{rec} and the matrix factorization loss Q_{MF} are defined as follows:

$$Q_{\text{rec}}(\boldsymbol{\theta}) = L(\mathbf{f}^u(\mathbf{g}^u(\mathbf{R},\mathbf{X}))) + L(\mathbf{f}^i(\mathbf{g}^i(\mathbf{R},\mathbf{Y}))),\ Q_{\text{MF}}(\mathbf{U},\mathbf{V}) = \sum_{j,k} \mathbb{1}(R_{jk}) ||R_{jk} - \mathbf{U}_{j,:}\mathbf{V}_{k,:}^T||^2 \tag{2}$$

where $\boldsymbol{\theta}$ contains all the parameters of the two autoencoders and $\mathbf{U}_{j,:}$ denotes the jth row of the matrix $\mathbf{U}$. The indicator function $\mathbb{1}(arg)$ returns 1 when $arg > 0$, and 0 otherwise.

We divide the objective function of Eq. (1) into three parts: 1) the reconstruction loss Q_{rec} trying to reconstruct the ratings and the side information of the users and items, 2) the matrix factorization (MF) loss Q_{MF} decomposing the rating matrix into user and item representations, which will be used for the prediction later, and 3) the third and fourth terms of Eq. (1) trying to keep the representations $\mathbf{U}$ and $\mathbf{V}$ in some distance from the neural representations. *We argue that these two terms play the role of regularizer, where they keep* $\mathbf{U}$ *and* $\mathbf{V}$ *from converging to the solution of MF.* The hyper-parameters λ_2 and λ_3 determine how far the $\mathbf{U}$ and $\mathbf{V}$ should be from the neural representations.

The main issue of the objective function in (1) is that the motivation behind using neural representation for the regularization purpose is unclear. Also, it is difficult to decide how far/close the neural and MF representations should be from each other, i.e., it is difficult to set the hyper-parameters λ_2 and λ_3.

Neural Representation for Prediction (NRP). We define the NRP framework that learns one set of user and item representations from the neural networks and uses them for the prediction directly, instead of using them as the regularizer. Similar to the previous works, our model contains two autoencoders, one for the users and one for the items. The difference is that the encoders' outputs are the only user/item representations in our model. Here is our objective function:

$$\min_{\boldsymbol{\theta}} Q_{rec}(\boldsymbol{\theta}) + \lambda_1 \sum_{j,k} \mathbb{1}(R_{jk}) ||R_{jk} - \mathbf{g}^u(\mathbf{R}_j, \mathbf{X}_j)^T \mathbf{g}^i(\mathbf{R}_k, \mathbf{Y}_k)||^2. \tag{3}$$

Our objective in Eq. (3) gives three advantages over the previous works: 1) the hyper-parameters λ_2 and λ_3, from Eq. (1), are removed, 2) the number of parameters decreased as we removed $\mathbf{U}$ and $\mathbf{V}$, which helps in faster training and saving memory, and 3) the network can be trained end-to-end, as there is no need to optimize over $\mathbf{U}$ and $\mathbf{V}$.

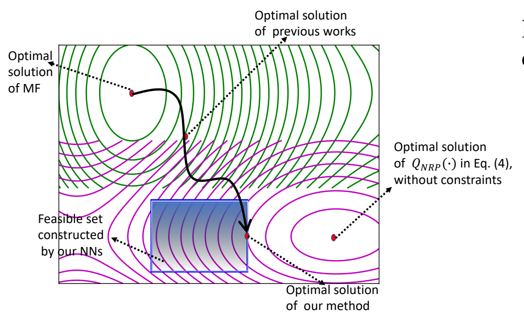
Optimal solution of previous works
Optimal solution of MF
Optimal solution of $Q_{NRP}(\cdot)$ in Eq. (4), without constraints
Feasible set constructed by our NNs
Optimal solution of our method

Figure 1: Visualization of the optimal solutions of different methods. The figure shows the contours over the users and items representations ($\mathbf{U}$ and $\mathbf{V}$).

Table 1: RMSE of NPR Compared with the baseline and state-of-the-art methods. "OM" means out of memory.

method	ml100k	ml1m	Amazon	Ichiba
MF [6]	0.940	0.892	1.153	1.00
Autorec [9]	0.921	0.889	2.19	2.47
NeuMF [5]	0.948	0.886	1.140	0.900
DHA [8]	0.939	0.865	OM	OM
NRP_{DHA}	0.926	0.855	1.135	OM
aSDAE [2]	0.946	0.879	OM	OM
$\text{NRP}_{\text{aSDAE}}$	0.910	0.877	1.24	OM
$\text{NRP}_{\text{direct}}$	**0.897**	**0.851**	**1.135**	**0.889**

We now analyze the objective of Eq. (3), compare it with the one in Eq. (1), and explain why the neural representations act as a regularizer in previous works. We rewrite our objective in (3) as:

$$\min_{\boldsymbol{\theta},\mathbf{U},\mathbf{V}} Q_{\text{NRP}}(\boldsymbol{\theta},\mathbf{U},\mathbf{V}) = Q_{\text{rec}}(\boldsymbol{\theta}) + \lambda_1 Q_{\text{MF}}(\mathbf{U},\mathbf{V}) \text{ s.t. } \quad \mathbf{U} = \mathbf{g}^u(\mathbf{R},\mathbf{X}) \quad \text{and} \quad \mathbf{V} = \mathbf{g}^i(\mathbf{R},\mathbf{Y}). \quad (4)$$

The objective functions in Equations (3) and (4) are equivalent, so we focus on comparing (4) with (1). We consider two special cases of the objective function in Eq. (1). First, consider the case where $\lambda_2 = \lambda_3 = 0$, which makes the last two terms 0. The first term Q_{rec} can also be removed since they do not contain the user/item representations $\mathbf{U}$ and $\mathbf{V}$. So only the MF term remains. The second case is when $\lambda_2 = \lambda_3 \to \infty$. We can theoretically show that in this case the two objective functions in (1) and (4) will be equivalent (i.e. they have the same optimal solution).

Fig. 1 shows a simple visualization of the objective functions and their optimal solutions. In this figure, the green contours correspond to the MF (by setting $\lambda_2 = \lambda_3 = 0$ in Eq. (1)) and the magenta contours correspond to $Q_{\text{NRP}}(\cdot)$, the main term of our objective function in Eq. (4). The feasible set, which satisfies our constraints in Eq. (4), has been shown by a blue rectangle. This feasible set contains the low-dimensional representations that can be created by the user and item encoders. The optimal solution of our objective function in Eq. (4) lies where the contour line of $Q_{\text{NRP}}(\cdot)$ with the smallest value intersects the feasible region.

By setting $\lambda_2 = \lambda_3 = 0$ and increasing it to $\lambda_2 = \lambda_3 \to \infty$, a path of solutions will be created, between the solution of the MF and our NRP autoencoder. The previous autoencoder methods use a fixed $\lambda_2 > 0$ and $\lambda_3 > 0$, so their optimal solution lies somewhere on the path. The smaller (larger) these hyper-parameters, the closer (farther away) the solution of the autoencoder-based methods will be to the MF's solution. *We believe the neural representations act as the regularizer in previous works since they are only used to keep* $\mathbf{U}$ *and* $\mathbf{V}$ *away from the MF's optimal solution.*

NRP with a direct structure. Here, our goal is to design a better network structure than autoencoders and combine it with the NRP structure. This (direct) structure is achieved by making two modifications to our autoencoder structure. First, we remove the decoders from the structure, which leads to saving around 50% of memory and faster optimization. Second, we use a set of fully connected layers to predict the final rating, instead of the dot product. This makes our model more expressive. The objective function can be achieved by removing $Q_{rec}(\boldsymbol{\theta})$ from Eq. (3) and using a multi-layer perceptron (instead of the dot product) to map the representations to the rating.

Experiments. We compare our NRP framework with the baselines and state-of-the-art methods on two public datasets (ml100k [3] and ml1m [3]) and two real-world datasets (Amazon [4] and Ichiba[1]). We report the root mean square error (RMSE) of each method in Table 1. DHA [8] and aSDAE [2], which are autoencoder-based methods optimizing Eq. (1), achieve lower RMSE than matrix factorization (MF). NRP_{DHA} and $\text{NRP}_{\text{aSDAE}}$, which apply the NRP framework to the encoder and decoder structures of DHA [8] and aSDAE [2], outperform DHA and aSDAE, respectively. *This means that neural representations are better for prediction than regularization.* Finally, $\text{NRP}_{\text{direct}}$ achieves the best results, with faster training and less memory usage compared to the autoecnoder-based methods.

[1]`https://rit.rakuten.co.jp/data_release/`

References

[1] R. Burke. Hybrid web recommender systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 377–408. Springer Berlin Heidelberg, 2007. doi: 10.1007/978-3-540-72079-9_12.

[2] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang. A hybrid collaborative filtering model with deep structure for recommender systems. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[3] F. M. Harper and J. A. Konstan. The MovieLens datasets. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, dec 2015. doi: 10.1145/2827872.

[4] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*. ACM Press, 2016. doi: 10.1145/2872427.2883037.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[6] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, aug 2009. doi: 10.1109/mc.2009.263.

[7] S. Li, J. Kawale, and Y. Fu. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM*. ACM Press, 2015. doi: 10.1145/2806416.2806527.

[8] T. Li, Y. Ma, J. Xu, B. Stenger, C. Liu, and Y. Hirate. Deep heterogeneous autoencoders for collaborative filtering. In *IEEE International Conference on Data Mining (ICDM)*, 2018.

[9] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie. AutoRec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. ACM Press, 2015. doi: 10.1145/2740908.2742726.

[10] F. Strub, R. Gaudel, and J. Mary. Hybrid recommender system based on autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems - DLRS 2016*. ACM Press, 2016. doi: 10.1145/2988450.2988456.

[11] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM Press, 2015. doi: 10.1145/2783258.2783273.

[12] S. Zhang, L. Yao, and X. Xu. AutoSVD++:an efficient hybrid collaborative filtering model via contractive auto-encoders. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 2017. doi: 10.1145/3077136.3080689.