Neural Representations in Hybrid Recommender Systems: Prediction versus Regularization

Ramin Raziperchikolaei Tianyu Li Young-joo Chung

Rakuten Institute of Technology, Rakuten, Inc.

Problem definition

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.

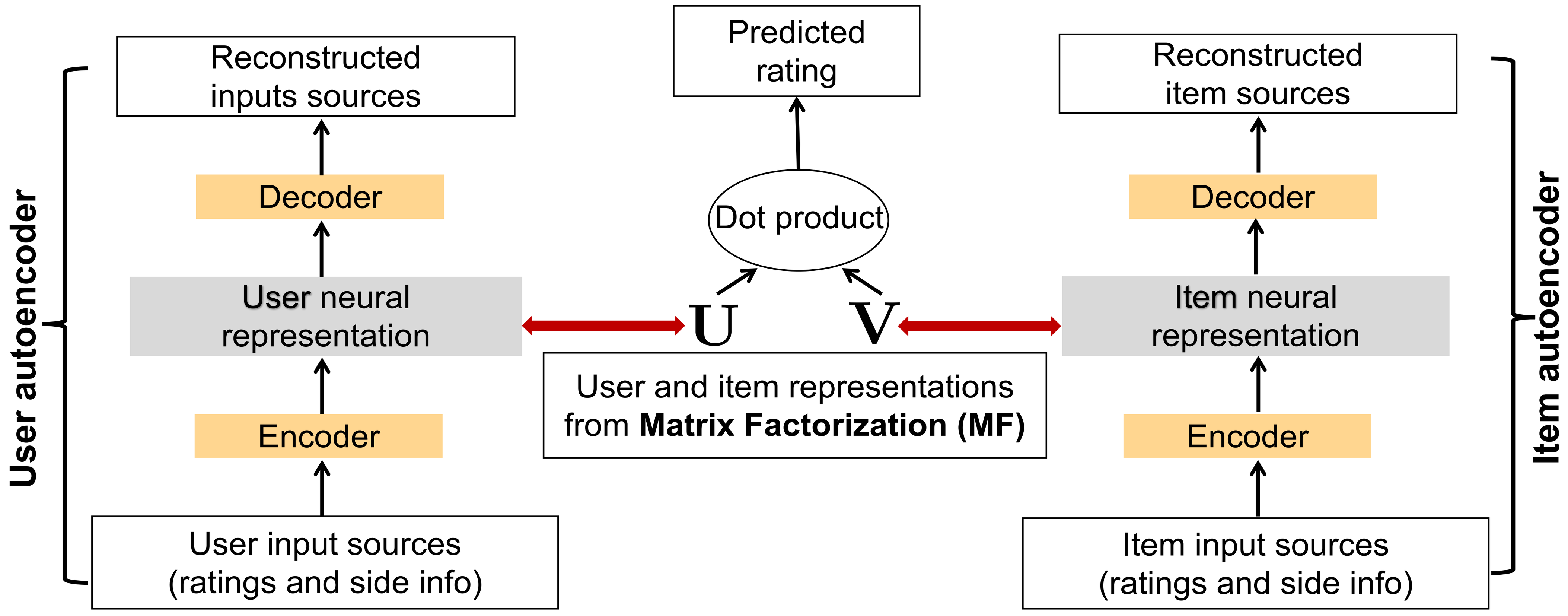
The goal of hybrid methods is to predict the unknown ratings using the known ratings and the user and item side information.

Autoencoder-based recommender systems

Autoencoder-based 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. They also define $\mathbf{U} \in \mathbb{R}^{m\times d}$ 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:

$$\min_{\mathbf{U},\mathbf{V},\boldsymbol{\theta}} L(\mathbf{f}^u(\mathbf{g}^u(\mathbf{R},\mathbf{X}))) + L(\mathbf{f}^i(\mathbf{g}^i(\mathbf{R},\mathbf{Y}))) + \lambda_1\textstyle\sum_{j,k}||R_{jk} - \mathbf{U}_{j,:}\mathbf{V}^T_{k,:}||^2$$
$$+\lambda_2||\mathbf{U} - \mathbf{g}^u(\mathbf{R},\mathbf{X})||^2 + \lambda_3||\mathbf{V} - \mathbf{g}^i(\mathbf{R},\mathbf{Y})||^2 + \text{reg. terms},$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_{f^u}, \boldsymbol{\theta}_{g^u}, \boldsymbol{\theta}_{f^i}, \boldsymbol{\theta}_{g^i}]$ contains all the parameters of the two autoencoders and $\mathbf{U}_{j,:}$ denotes the ith row of the matrix $\mathbf{U}$. The rating of the user j on item k is approximated by the dot product of the $\mathbf{U}_{j,:}$ and $\mathbf{V}_{k,:}$.



Issues of the previous autoencoder-based methods:

- The motivation behind using neural representation for the regularization is unclear. How far/close the neural and MF representations should be from each other?
- Optimization is difficult and time-consuming.
- The dot product to predict ratings from representations $\mathbf{U}$ and $\mathbf{V}$ might not be sufficient to combine the two representations.

Our approach: Neural Representation for Prediction (NRP)

We introduce the **N**eural **R**epresentation for **P**rediction (**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:

$$\min_{\boldsymbol{\theta}} L(\mathbf{f}^u(\mathbf{g}^u(\mathbf{R},\mathbf{X}))) + L(\mathbf{f}^i(\mathbf{g}^i(\mathbf{R},\mathbf{Y}))) + \lambda_1 \sum_{j,k} ||R_{jk} - \mathbf{g}^u(\mathbf{R}_{j,:},\mathbf{X}_{j,:})^T\mathbf{g}^i(\mathbf{R}_{:,k},\mathbf{Y}_{k,:})||^2.$$

Our objective function gives three advantages over the previous works:

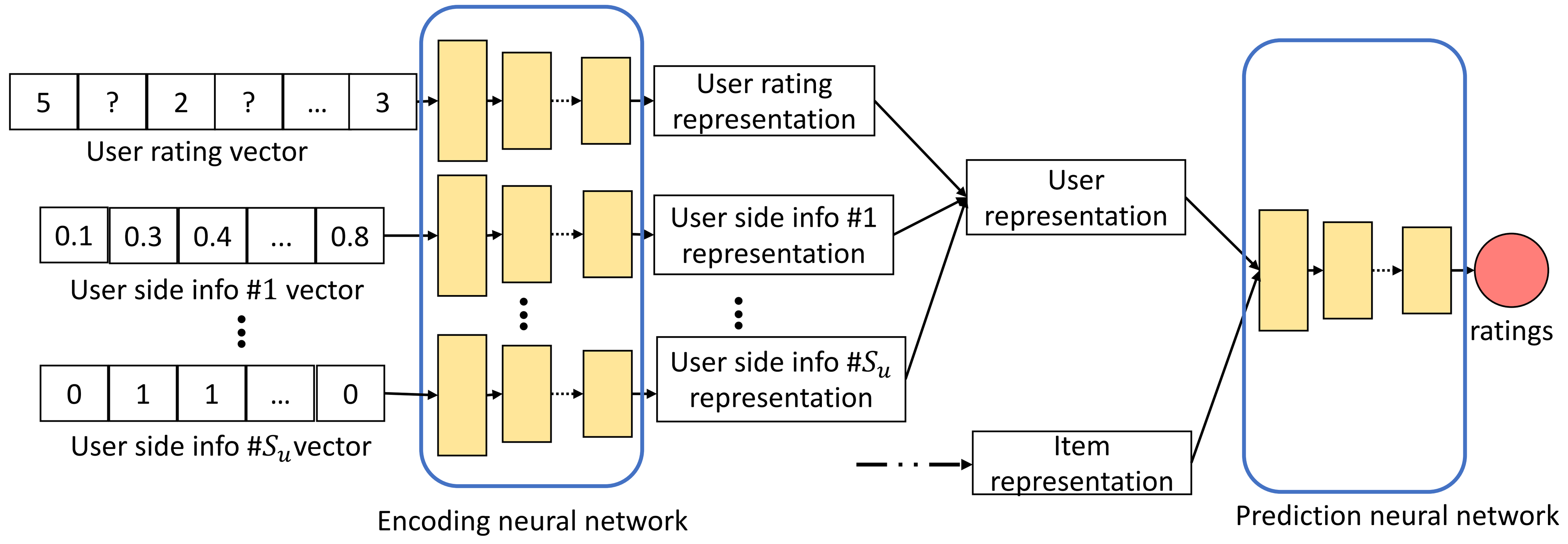
- The hyper-parameters λ_2 and λ_3 are removed.
- The number of parameters decreased as we removed $\mathbf{U}$ and $\mathbf{V}$, which helps in faster training and saving memory.
- Our model is trained end-to-end, as there is no need to optimize over $\mathbf{U}$ and $\mathbf{V}$.

By setting $\lambda_2 = \lambda_3 = 0$ and increasing it to $\lambda_2 = \lambda_3 \rightarrow \infty$, 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.

NRP with a direct structure

Our direct structure is achieved by:

- Removing the decoders from the structure, which leads to saving around 50% of memory and faster optimization.
- Using a set of fully connected layers to predict the final rating, instead of the dot product. This makes our model more expressive.



Experiments

We use the follwoing four datasets in our experiments:

	# of users	# of items	sparsity
ml100k	1 000	1 600	94%
ml1m	6 000	4 000	96%
Amazon Review	86 400	108 500	99.994%
Ichiba	324 000	294 000	99.84%

In the following table, we compare our proposed NRP framework, trained with the autoencoder and direct structures, versus MF and autoencoder-based methods on ml1m datasets.

Our framework combined with the direct structure achieves better prediction results, faster training, and less memory usage compared to the autoencoder-based methods.

	ml1m			
method	RMSE	precision	# params.	time
MF	0.892 ± 0.004	68.2% ± 0.3	(0, 1M)	45s
DHA	0.865 ± 0.001	69.3% ± 0.2	(44M, 1M)	1 097s
NRP$_{\text{DHA}}$	0.855 ± 0.002	69.6% ± 0.2	(44M, 0)	1 027s
aSDAE	0.879 ± 0.005	69.0% ± 0.1	(66M, 1M)	1 155s
NRP$_{\text{aSDAE}}$	0.877 ± 0.008	68.5% ± 0.4	(66M, 0)	1 055s
NRP$_{\text{direct}}$	**0.851** ± **0.001**	**70.0**% ± **0.1**	(22M, 0)	640s

In the following table, we compare RMSE of our NRP framework with the hybrid and collaborative filtering methods.

Our approach outperforms the rest of the methods.

method	ml100k	ml1m	Amazon	Ichiba
MF	0.940	0.892	1.153	1.00
Autorec	0.921	0.889	2.19	2.47
NeuMF	0.948	0.886	1.140	0.900
DSSM	0.934	0.941	NA	0.913
DHA	0.939	0.865	OM	OM
NRP$_{\text{DHA}}$	0.926	0.855	1.135	OM
aSDAE	0.946	0.879	OM	OM
NRP$_{\text{aSDAE}}$	0.910	0.877	1.24	OM
HIRE	0.930	0.861	OM	OM
NRP$_{\text{direct}}$	**0.897**	**0.851**	**1.135**	**0.889**