

RLT: Residual-Loop Training in Collaborative Filtering for Combining Factorization and Global-Local Neighborhood

Lei Li^{1,2}, Weike Pan^{1*}, Li Chen², and Zhong Ming^{1*}

lilei1995eli@gmail.com, panweike@szu.edu.cn, lichen@comp.hkbu.edu.hk, mingz@szu.edu.cn

¹College of Computer Science and Software Engineering
Shenzhen University, Shenzhen, China

²Department of Computer Science
Hong Kong Baptist University, Hong Kong, China

Problem Definition

	i1	i2	i3	i4	i5	i6		
u1	?	3	?	?	?	?		
u2	?	?	3	1	2	?		
u3	?	★★★★★				1	user	
u4	?	★★★★★				5		
u5	?	5	?	5	?	5		
u6	4	?	?	1	?	?		
								item

Rating Prediction

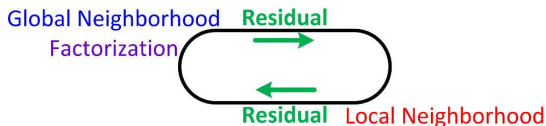
- Input: A set of (user, item, rating) triples as training data denoted by $\mathcal{R} = \{(u, i, r_{ui})\}$, where r_{ui} is the numerical rating assigned by user u to item i .
- Goal: Estimate the preference of user u to item j , i.e., \hat{r}_{uj} , for each record in the test data $\mathcal{R}^{te} = \{(u, j, r_{uj})\}$.

Limitations of Related Work

Traditional pipelined residual training paradigm may not be able to fully exploit the merits of factorization- and neighborhood-based methods.

- 1 There are two different types of neighborhood, i.e., global neighborhood in FISM and SVD++, and local neighborhood in ICF, but most residual training approaches ignore the global neighborhood.
- 2 Combining the factorization-based method and neighborhood-based method in a pipelined residual chain may not be the best because the one-time interaction between the two methods may not be sufficient.

Overall of Our Solution



Residual-Loop Training (RLT): a new residual training paradigm, which aims to fully exploit the complementarity of factorization, global neighborhood and local neighborhood in one single algorithm.

Advantages of Our Solution

- 1 We recognize the difference between global neighborhood and local neighborhood in the context of residual training.
- 2 We propose to combine factorization-, global neighborhood-, and local neighborhood-based methods by residual training.
- 3 We propose a new residual training paradigm called residual-loop training (RLT).

Notations

Table: Some notations and explanations.

u	user ID
i, i', j	item ID
r_{ui}	rating of user u to item i
$\mathcal{R} = \{(u, i, r_{ui})\}$	rating records of training data
\mathcal{U}_i	users who rate item i
\mathcal{I}_u	items rated by user u
\mathcal{N}_i	nearest neighbors of item i
$\mu \in \mathbb{R}$	global average rating value
$b_u \in \mathbb{R}$	user bias
$b_j \in \mathbb{R}$	item bias
$d \in \mathbb{R}$	number of latent dimensions
$U_{u.} \in \mathbb{R}^{1 \times d}$	user-specific latent feature vector
$V_{i.}, W_{j.} \in \mathbb{R}^{1 \times d}$	item-specific latent feature vector
$\mathcal{R}^{te} = \{(u, j, r_{uj})\}$	rating records of test data
\hat{r}_{ui}	predicted rating of user u to item i
λ	tradeoff parameter
T	iteration number in the algorithm

Factorization-based Method

Probabilistic matrix factorization (PMF) is a factorization-based method for rating prediction in collaborative filtering. Specifically, the prediction rule of the rating assigned by user u to item i is as follows,

$$\hat{r}_{ui}^F = \mu + b_u + b_i + U_u \cdot V_i^T, \quad (1)$$

where μ , b_u and b_i are the global average, the user bias, and the item bias, respectively, and $U_u \in \mathbb{R}^{1 \times d}$ and $V_i \in \mathbb{R}^{1 \times d}$ are the user-specific latent feature vector and the item-specific latent feature vector, respectively.

Local Neighborhood-based Method

Item-oriented collaborative filtering (ICF) is a neighborhood-based method for preference estimation in recommendation. The estimated preference of user u to item i can be written as follows,

$$\hat{r}_{ui}^{\text{Ne}} = \sum_{i' \in \mathcal{I}_u \cap \mathcal{N}_i} \bar{s}_{i'i} r_{ui'}, \quad (2)$$

where $\bar{s}_{i'i} = s_{i'i} / \sum_{i' \in \mathcal{I}_u \cap \mathcal{N}_i} s_{i'i}$ is the normalized similarity with $s_{i'i} = |\mathcal{U}_{i'} \cap \mathcal{U}_i| / |\mathcal{U}_{i'} \cup \mathcal{U}_i|$ as the Jaccard index between item i' and item i .

- \mathcal{N}_i is a set of locally nearest neighboring items of item i , i.e., **their similarities are predefined without global propagation** among the users, thus we call it a local neighborhood-based method.

Global Neighborhood-based Method

The **similarity** in Eq.(2) may also be **learned** from the data instead of being calculated, e.g., in asymmetric factor model (AFM), the prediction rule of user u to item i is as follows,

$$\hat{r}_{ui}^{\text{Ng}} = \sum_{i' \in \mathcal{I}_u \setminus \{i\}} \bar{p}_{i'i}, \quad (3)$$

where $\bar{p}_{i'i} = W_{i'} \cdot V_i / \sqrt{|\mathcal{I}_u \setminus \{i\}|}$.

- 1 Two items without common users may still be well connected via the learned latent factors.
- 2 The prediction rule in Eq.(3) does not restrict to a local neighborhood set \mathcal{N}_i as that in Eq.(2).

We thus call AMF with the prediction rule in Eq.(3) a global neighborhood-based method.

Factorization- and Global Neighborhood-based Method

Matrix factorization with implicit feedback (SVD++) integrates the prediction rules of a factorization-based method and a global neighborhood-based method,

$$\begin{aligned}
 \hat{r}_{ui}^{F-Ng} &= \mu + b_u + b_i + U_u \cdot V_i^T + \sum_{i' \in \mathcal{I}_u \setminus \{i\}} \bar{p}_{i'i}, \\
 &= \hat{r}_{ui}^F + \hat{r}_{ui}^{Ng},
 \end{aligned} \tag{4}$$

from which we can see that SVD++ is **a generalized factorization model** that inherits the merits of both factorization- and global neighborhood-based methods.

Residual Training

Residual training (RT) is **an alternative approach** to combining a **factorization-based method** and a **neighborhood-based method**. Specifically, a factorization-based model is built using the training data, and a predicted rating \hat{r}_{ui}^F for each $(u, i, r_{ui}) \in \mathcal{R}$ can then be obtained, based on which a neighborhood-based method is developed using $\sum_{i' \in \mathcal{I}_u \cap \mathcal{N}_i} \bar{s}_{i'i} r_{ui'}^{res}$, where $r_{ui'}^{res} = r_{ui'} - \hat{r}_{ui'}^F$ is the residual. The learning procedure can be represented as follows,

$$\hat{r}_{ui}^F \rightarrow \hat{r}_{ui}^{N_\ell}. \quad (5)$$

The final prediction rule is then the summation of \hat{r}_{ui}^F and $\hat{r}_{ui}^{N_\ell}$, i.e., $\hat{r}_{ui}^F + \hat{r}_{ui}^{N_\ell}$.

Differences between SVD++ and RT

The main differences between SVD++ and RT are:

- 1 SVD++ is **an integrative method** with one single prediction rule, while RT is **a two-step approach** with two separate prediction rules.
- 2 SVD++ exploits **factorization and global neighborhood**, while RT makes use of **factorization and local neighborhood**.

Residual-Loop Training (1/3)

In order to fully exploit the **complementarity** of factorization, global neighborhood and local neighborhood, we propose a new residual training paradigm called residual-loop training (RLT), which is depicted as follows,

$$\hat{r}_{ui}^{F-Ng} \rightarrow \hat{r}_{ui}^{N\ell} \rightarrow \hat{r}_{ui}^{F-Ng} \quad (6)$$

where \hat{r}_{ui}^{F-Ng} is from Eq.(4) and $\hat{r}_{ui}^{N\ell}$ is from Eq.(2).

Residual-Loop Training (2/3)

- 1 For the first \hat{r}_{ui}^{F-Ng} in Eq.(6), we aim to exploit both **factorization and global neighborhood**. The interaction between the factorization-based method and the global neighborhood-based method is richer in such an integrative method than that in two separate steps of RT.
- 2 For \hat{r}_{ui}^{Ng} , we aim to boost the performance via **local neighborhood**, i.e., explicitly combining factorization, global neighborhood and local neighborhood for rating prediction in a residual-training manner.
- 3 For the second \hat{r}_{ui}^{F-Ng} , we aim to further **capture the remaining effects** related to users' preferences that have not been modeled by the previous two methods yet.

Residual-Loop Training (3/3)

Input: Users' rating records $\mathcal{R} = \{(u, i, r_{ui})\}$.

Output: Predicted preference of each record in the test data, i.e., $\hat{r}_{uj}, (u, j) \in \mathcal{R}^{te}$.

Task 1. Conduct factorization- and global neighborhood-based preference learning (i.e., SVD++), and estimate the preference of each record in the training data \hat{r}_{ui}^{F-Ng} and the preference of each record in the test data \hat{r}_{uj}^{F-Ng} .

Task 2. Conduct local neighborhood-based preference learning (i.e., ICF) on the residual $r_{ui} - \hat{r}_{ui}^{F-Ng}$, and estimate the preference of each record in the training data $\hat{r}_{ui}^{N\ell}$ and the preference of each record in test data $\hat{r}_{uj}^{N\ell}$.

Task 3. Conduct factorization- and global neighborhood-based preference learning again (i.e., SVD++) on the residual $r_{ui} - \hat{r}_{ui}^{F-Ng} - \hat{r}_{ui}^{N\ell}$, and estimate the preference of each record in the test data $\hat{r}_{uj}^{F-Ng'}$. Finally, the prediction of each record in the test data is obtained $\hat{r}_{uj} = \hat{r}_{uj}^{F-Ng} + \hat{r}_{uj}^{N\ell} + \hat{r}_{uj}^{F-Ng'}$.

Figure: The algorithm of residual-loop training (RLT).

Datasets and Evaluation Metric

- We conduct extensive experiments on three public datasets, including MovieLens 100K ([ML100K](#)), MovieLens 1M ([ML1M](#)) and MovieLens 10M ([ML10M](#))¹.
- Each dataset is divided into training and test sets with the proportion of **80% and 20%** respectively, and the splitting procedure is repeated for five times for five-fold cross validation.
- We adopt the commonly used root mean square error (**RMSE**) in our performance evaluation, and report the average result from five-time evaluation.

¹<http://grouplens.org/datasets/movielens/>

Baselines

- Item-oriented collaborative filtering (ICF) with Jaccard index as the similarity measurement.
- Probabilistic matrix factorization (PMF).
- Hybrid collaborative filtering (HCF) that averages the predictions of ICF and PMF, i.e., $\hat{r}_{ui} = (\hat{r}_{ui}^{ICF} + \hat{r}_{ui}^{PMF})/2$.
- Singular value decomposition with implicit feedback (SVD++).
- Residual training (RT) with PMF and ICF as two dependent components in a sequential manner.

Parameter Configurations

- For all factorization-based methods, we fix the number of latent dimensions as $d = 20$, the learning rate $\gamma = 0.01$, the iteration number as $T = 50$, and search the value of tradeoff parameters from $\{0.001, 0.01, 0.1\}$.
- For neighborhood-based methods, we take top-20 items from $\mathcal{I}_u \cap \mathcal{N}_i$ with highest Jaccard index as the neighbors. Notice that when $|\mathcal{I}_u \cap \mathcal{N}_i| < 20$, we use all items from $\mathcal{I}_u \cap \mathcal{N}_i$.

Main Results (1/4)

Table: Recommendation performance of item-oriented collaborative filtering (ICF), probabilistic matrix factorization (PMF), hybrid recommendation combining ICF and PMF (HCF), SVD++, residual training (RT) and our residual-loop training (RLT). The significantly best results are marked in bold ($p < 0.01$). The values of the tradeoff parameter λ are also included for reproducibility.

	ML100K	ML1M	ML10M
ICF	0.9537 \pm 0.0038	0.9093 \pm 0.0021	0.8683 \pm 0.0012
PMF	0.9441 \pm 0.0038 ($\lambda = 0.01$)	0.8838 \pm 0.0023 ($\lambda = 0.001$)	0.7911 \pm 0.0005 ($\lambda = 0.01$)
HCF	0.9242 \pm 0.0032 ($\lambda = 0.01$)	0.8739 \pm 0.0023 ($\lambda = 0.001$)	0.8052 \pm 0.0007 ($\lambda = 0.01$)
SVD++	0.9246 \pm 0.0031 ($\lambda = 0.001$)	0.8515 \pm 0.0018 ($\lambda = 0.001$)	0.7873 \pm 0.0007 ($\lambda = 0.01$)
RT	0.9145 \pm 0.0041 ($\lambda = 0.001$)	0.8567 \pm 0.0021 ($\lambda = 0.001$)	0.7847 \pm 0.0008 ($\lambda = 0.01$)
RLT	0.8968 \pm 0.0040 ($\lambda = 0.001$) ($\lambda = 0.001$)	0.8385 \pm 0.0016 ($\lambda = 0.001$) ($\lambda = 0.001$)	0.7812 \pm 0.0007 ($\lambda = 0.01$) ($\lambda = 0.01$)

Main Results (2/4)

Observations

- Our RLT predicts the users' preferences **significantly more accurately** than all other baseline methods, which clearly shows the advantage of our residual-loop training paradigm.
- For the performance of SVD++ and RT, we can see that their performance results are **very close** though the former exploits factorization and global neighborhood in an integrative way, and the latter exploits the factorization and local neighborhood in a pipelined manner, which also motivates us to further exploit the complementarity of factorization, global neighborhood, and local neighborhood.

Main Results (3/4)

We further study the performance of each task in our RLT.

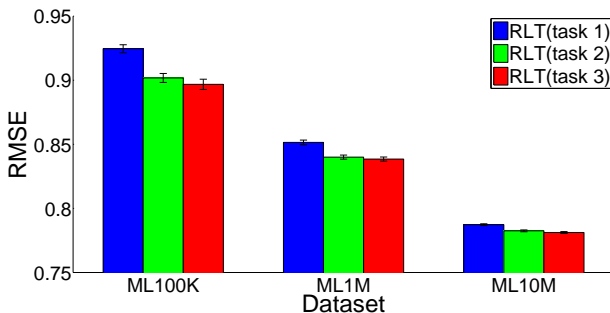


Figure: Recommendation performance of three tasks in RLT, i.e., task 1 is SVD++, task 2 is ICF, and task 3 is SVD++ again.

Main Results (4/4)

Observations

- The performance **improves in each subsequent task**, e.g., “from SVD++ to ICF” and “from ICF to SVD++”, which shows the effectiveness of our residual-training mechanism that links factorization- and global-local neighborhood-based methods.
- The improvement “from SVD++ to ICF” is much **larger** than that “from ICF to SVD++”, which implies that the second task is very useful while the third task is only marginally useful. This can be interpreted by the fact that the factorization and global-local neighborhood are somehow already well exploited in “SVD++ to ICF”. Notice that although the further improvement in the third task of “from ICF to SVD++” is small, the improvement is still **statistically significant**.

Conclusions

- We design a new residual training paradigm called **residual-loop training** (RLT), which aims to combine factorization, global neighborhood and local neighborhood in one single algorithm so as to fully exploit their complementarity.
- Experimental results on three public datasets show the **significantly better** performance of our RLT than several state-of-the-art factorization- and neighborhood-based methods.

Thank you!

- We thank the anonymous reviewers for their expert and constructive comments and suggestions.
- We thank the support of National Natural Science Foundation of China Nos. 61502307, 61672358 and 61272365, Hong Kong RGC under the project RGC/HKBU12200415, and Natural Science Foundation of Guangdong Province Nos. 2014A030310268 and 2016A030313038.