Large-scale Distributive Matrix Collaborative Filtering for Recommender System

Farhan Ullah†
School of Computer Engineering and Science, Shanghai University, Shanghai, China, 200444
farhan.marwat@gmail.com

Bofeng Zhang
School of Computer Engineering and Science, Shanghai University, Shanghai, China, 200444
bfzhang@shu.edu.cn

Guobing Zou
School of Computer Engineering and Science, Shanghai University, Shanghai, China, 200444
guobingzou@gmail.com

Irfan Ullah
Department of Electrical Engineering, Chulalongkorn University, Bangkok, Thailand
irfan.ee@cqu.edu.cn

Ali Mustafa Qamar
Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia
Al.khan@qu.edu.sa

Durr-e-Nayab
Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan
nayab.khan@uetpeshawar.edu.pk

ABSTRACT
In the modern world, people face an explosion of information and difficulty finding the right choice for their interests. Nowadays, people prefer online shopping for their needs. Recently, the recommender system has become one of the key technology for the online purchasing system. The collaborative filtering technique has been extensively applied for the Recommender Systems. However, collaborative filtering is suffering from data sparsity, cold start problems, and inaccuracy problems. To overcome these problems, we propose a novel approach of the Matrix Distributive collaborative filtering with ensemble integration. The experimental results illustrate the increase in performance against the existing methods.

CCS CONCEPTS
Matrix Factorization, Distributive Matrix, Collaborative Filtering, Ensemble Integration, Recommender System

KEYWORDS
Collaborative Filtering, Ensemble approach, Matrix Factorization, Distributive Matrix.

ACM Reference format:
Farhan Ullah, Bofeng Zhang, Guobing Zou, Irfan Ullah, Ali Mustafa Qamar and Durr-e-Nayab. 2020. Large-scale Distributive Matrix Collaborative Filtering for Recommender System. In *Proceedings of 2020 International Conference on Computing, Networks and Internet of Things (CNIOT' 20). Sanya, China, 5 pages.* https://doi.org/10.1145/3398329.3398360

1. Introduction
The Collaborative Filtering (CF) rating prediction has been a hot research problem in the recommendation system recently [1]. Many CF methods have been introduced recently [2]. The Collaborative Filtering technique takes users and items' rating matrix to build the recommendation system. The rating of items usually suggests the interests of users for such items. While other features of users, for example, age, gender, and location, are hard to navigate because of privacy issues, the ratings have the primary role in navigating the user's interest and recommend more relevant items for such users based on the user's past rating history.
Many models have surfaced to tackle this issue, such as memory-based methods [3], model-based algorithms [4]. The memory-based model for collaborative filtering predicts the rating of users and items by using the weighted average of users and items' rating. Some of the collaborative filtering methods use the discrete rating instead of continuous ratings. While other methods focus on continuous rating values, the evaluation methods of collaborative filtering are Root Mean Square Error (RMSE) [5].
The Netflix prize competition is a significant contest for CF techniques. This contest boosts up the progress and efficiency of the techniques, and many researchers presented their work in this contest in October 2006 [6]. Netflix issued a large dataset of users, items, and movie ratings, which consisted of over 480 thousand randomly selected customers and 18 thousand movie items. The primary purpose of this competition is to develop such algorithms that can beat the accuracy of their current Cine-match to some extent. In the Netflix prize competition, RMSE is adopted for performance evaluation [7]. However, one of the methods, the Matrix Factorization technique got impressive results. Matrix Factorization can easily find latent factors [8]. This technique

†Corresponding Author: farhan.marwat@gmail.com

CNIOT2020, April 24–26, 2020, Sanya, China

ACM ISBN 978-1-4503-7771-3/20/04…$15.00
https://doi.org/10.1145/3398329.3398360

divides the original matrix in such a way that if we multiply them back, we get approximately similar results as an original matrix.

$$\boldsymbol{R} = \boldsymbol{p^t}.\boldsymbol{q} \quad \textbf{(1)}$$

where R represents the original matrix and p and q are the latent factors and are row vectors.

After obtaining successful results from this novel technique, researchers modified this method and came up with more improvements and efficient Matrix Factorization techniques such as Singular Value Decomposition (SVD)++ [9]. We adopt the same intuition and bring a novel technique called Distributive matrix collaborative filtering for recommender systems. By conducting extensive experiments, our model has shown improvement in RMSE as compared to the existing ones.

2. Literature Review

CF [10], [11] is a very effective method for recommending similar items in the field of recommender systems. The CF technique predicts users' interest in the items from other most similar users' interest items and recommends the nearest items to those users. The CF problem is formulated as (U, I, R), where:

- U is a set of M users with the values $\{u_1, u_2, \ldots, u_m\}$
- I is a set of N items with the values $\{i_1, i_2, \ldots, i_n\}$
- R a rating matrix, which consists of users' historical ratings record.

The CF predicted value estimates the interest of user U to item I. It implies that the higher the value, the stronger the interest of users for the item. The CF technique suffers from the primary challenge of rating accuracy prediction to unknown ratings in the matrix, in general, which is a very sparse rating matrix. There are several CF techniques introduced in a couple of years; for example, K-Nearest Neighbor (K-NN) based models [12], [13] and Matrix Factorization (MF) models [14], [15]. An ensemble model is a well-known machine learning approach that combines the results of different machine learning models and presents them as a single and more effective model to improve the predictive results. In contrast, the hybrid method of recommendation system [16] combines the content-based model and CF model. Recently, more sophisticated methods have been proposed to enhance the accuracy of collaborative filtering techniques. The CF enhancement methods include various Matrix Factorization techniques [17], applying the implicit dataset in the model training [18], applying various similarity measures [19], and applying Gradient Descent-based techniques [11]. In another paper [20], the CF rating sparsity problem has been resolved by assigning the artificial dataset in the rating matrix of the collaborative filtering. Several machine learning models have been applied for the evaluation, including ensembles. The authors in [21], have presented three different Matrix Factorization techniques, which differ in the number of parameters, by improving the Matrix Factorization optimization problem. They achieved improved results by using an ensemble model, which was based on a simple average of the three Matrix Factorization techniques. In another paper [22], several K-NN models have been combined to improve Mean Absolute Error (MAE) of the recommendation task. The combination strategy was based on the User-based and Item-based CF technique. They also applied a bagging machine learning approach to user and item similarities. Ho et al. [23] modified the existing AdaBoost version, which showed improvement in the RMSE measure for the Matrix Factorization model. The authors also demonstrated the effectiveness by adding more models to the ensemble to reduce the RMSE; they showed improvement with a maximum number of 10 models. In another paper [24], the author introduced the heterogeneous ensemble model with a combination of 5 different CF methods. The result of a combined model was higher as compared to base models. In this paper, we introduced a novel framework of the ensemble method by adopting the efficient Matrix Factorization technique for applying on a large-scale CF technique. We integrate the homogeneous ensemble approach based on similar Matrix Factorization models. Our framework has shown effectiveness against the baseline and state-of-the-art CF techniques. Our model also exhibits improvement in RMSE and MAE against all the state-of-the-art methods in collaborative filtering. In [25], the author presented an effective distributed machine learning method for machine learning, but their work is different from our presented work in the paper. In paper [26], the author proposed the parallel Matrix Factorization technique for rating prediction. In [27], the authors modified the existing SGD method for fast parallel Matrix Factorization, but their application was not applicable. Li et al. [28] introduced a novel Matrix Factorization method called MSGD, based on multi-GPU. In [29], the author enhances the Matrix Factorization technique with memory optimization and computing.

3. Proposed Methodology

In this section, we will discuss our proposed method.

3.1 Matrix Factorization

Matrix Factorization technique [30], belongs to the class of collaborative filtering-based techniques. The MF method is often applied in recommender systems to find the latent factors in a matrix. The Matrix Factorization technique converts the original matrix into the product of two lower-dimensional matrixes, as shown in Figure 1. The MF technique got popularity during the Netflix prize competition. In this competition, it got better results in predicting the ratings of the users and items of Movie Lens dataset. Low-rank Matrix Factorization is one of the most famous and is frequently used in recommender systems and other domains of machine learning. To define MF, suppose we have a set of users (U) and a set of items (I). Let R be the product of U and I matrices which contain all the ratings that the users have given to the items. Our goal is to find two matrices, $p\ ((u) * k)\ and\ ((i) * k)$ such that the product of these matrices is equal to matrix R as shown in Eq. 2.

$$\boldsymbol{R} = \boldsymbol{p^t}.\boldsymbol{q} = \boldsymbol{\widehat{R}} \quad \textbf{(2)}$$

where $\boldsymbol{R}$ is the actual rating, $\boldsymbol{p^t}$ denotes the users' matrix, $\boldsymbol{q}$ represents the items matrix, and $\boldsymbol{\widehat{R}}$ is the predicted ratings.
The model of the Matrix Factorization maps both users and items in the latent space of dimensionality. The interaction between user and item is formed as an inner product in the space. Thus, each item is connected to the $q \epsilon R$ vector, and each user is connected to the $p_u \epsilon R$ vector. The result is the multiplication of q and p_u, which gives the interaction between item I and user U. This approximates the user ratings given to items, which are denoted by $r_{u,i}$ as shown in Eq. 3.

$$\boldsymbol{r_{u,i}} = \boldsymbol{q_i^t}.\boldsymbol{p_u} \quad \textbf{(3)}$$

Now the system needs to minimize the regularized square error to learn the vectors associated with factors such as p_u and q_i as described in Eq. 4.

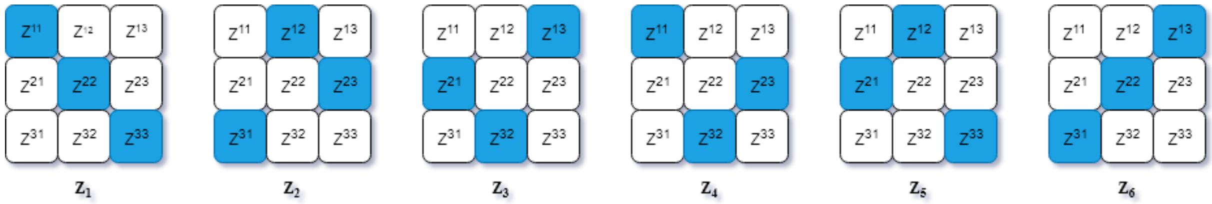

Figure 2. Example of 3x3 strata of Distributive Matrix Factorization

$$\min_{Q,P} \sum_{(u,i)\in k} (\boldsymbol{r}_{u,i} - \boldsymbol{q}_i^T \boldsymbol{p}_u)^2 + \boldsymbol{\lambda}(||\boldsymbol{q}_i||^2 + ||\boldsymbol{p}_u||^2) \quad (4)$$

k is the set of the (U, I) pairs for which $r_{u,i}$ is known as the training set. Lambda (λ) controls the measure of regularization and is generally determined by cross-validation.

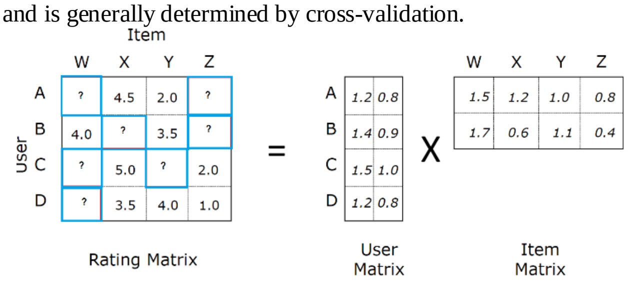

Figure 1. An example of Matrix Factorization

3.2 Distributive Stochastic Gradient Descent (Dsgd)

We first discuss the Stochastic Gradient Descent (SGD) in Matrix Factorization [8]. The whole process behind SGD is to select $r_{u,i}$ in R matrix before finding the factor p_u from the users' factor matrix P. Similarly, the factor matrix q_i is determined from the Q factor matrix. During the last step, it calculates the predicted score $p_u q_i^t$ and keeps updating the parameters continuously using Eq. 5-6.

$$\boldsymbol{p}_u \leftarrow \boldsymbol{p}_u + \boldsymbol{\gamma}(\boldsymbol{e}_{u,i}\boldsymbol{q}_i - \boldsymbol{\lambda}\boldsymbol{p}_u) \quad (5)$$

$$\boldsymbol{q}_i \leftarrow \boldsymbol{q}_i + \boldsymbol{\gamma}(\boldsymbol{e}_{u,i}\boldsymbol{p}_u - \boldsymbol{\lambda}\boldsymbol{q}_i) \quad (6)$$

The SGD algorithm is a sequential method to update the factor matrices and parameters. The logic behind the Distributive Stochastic Gradient Descent (DSGD) is to switch the order of SGD steps without affecting the output. This procedure allows to run SGD in parallel with interchangeable sets of rating matrix [31]. DSGD divides the rating matrix Z into a set of subsets of matrix z, called as *strata*. Individual stratum consists of d interchangeable subsets of Z as shown in Figure 2. The parameter of the parallelism should be taken equal to or greater than the total number of processing tasks. In the next step, they start permutation of the rows and columns of the matrix r and then create $d \times d$ blocks. Afterwards, they block the factor matrices q and p. This process ensures the same number of training points in each block. After the iteration of $u_1, u_2, \ldots, u_d$ of $1, 2, 3, \ldots, d$, the stratum becomes $z_s = z^{1u_1} \cup z^{2u_2} \cup \ldots \cup z^{du_d}$ where the substratum $z^{u,i}$ is the same as a block $r^{u,i}$. It should be noted that overlapping is possible when parameter d is greater than 2.

3.3 Ensemble Integration

Ensemble learning techniques have been very useful in manipulating the large-scale and small-scale datasets. They are also widely employed for classification and regression problems in computer vision and various machine learning problems. In this paper, we have only used the rating information of users and items, hence we present it as a regression problem. We employ the Ensemble technique to address the regression problem, and also, we integrate the Distributive Matrix with the ensemble technique for further enhancing the performance of the Distributive Matrix Factorization. We used the same nature of the Distributive Matrix Factorization technique in the ensemble algorithm, unlike the heterogeneous approach of the ensemble. Since we have a regression problem, so we take a weighted average of the two same nature of distributive matrices in the ensemble technique.

4. Experiments and Results

We performed several experiments to show the effectiveness of our proposed model and framework. We also do some extended experiments to examine the performance of different models of Matrix Factorization, such as Alternating Least Square (*ALS*), user-item collaborative filtering.

4.1 Dataset

We evaluate our models on a widely used Movie lens data set [32] in recommender systems (ML4M). These datasets are publicly available at www.grouplens.org. We do not need to process the Movie lens datasets because it is already filtered. The details of the three data sets are given in Table 1.

Table 1. Details of Movie Lens dataset

Statistics	ML4M
Number of users	28667
Number of items	44270
Number of ratings	4000054
Rating density	0.23657

4.2 Evaluation

There are several types of measures for the evaluation of recommender systems. One of the most common measure is the Root Mean Square Error (RMSE). It calculates the average of all squared differences between the actual and predicted rating values and then continues to calculate the square root of the results. Therefore, larger errors can significantly affect the RMSE rating value. Eq. 7 shows the RMSE between the original rating values and the predicted rating values:

$$\boldsymbol{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{d_i} - \widehat{\boldsymbol{d_\iota}})^2} \qquad \textbf{(7)}$$

where d_i is the value of the actual rating, $\widehat{d_\iota}$ is the value of the predicted rating and n represents the total number of ratings.

4.3 Implementation Details

In this subsection, we discuss the parameters used in the proposed model. When training our models, we composed the *Distributive Matrix with Ensemble* (*DMWE*) model with a total of two base learners and the composite learner parameters with a weighted average of 1.0. For Distributive Matrix (DM), we initialized model parameters with a maximum rating scale of 5 and the number of factors as 12. Similarly, user and item regularization is set to 0.015 whereas the number of iterations was fixed as 30, regularization as 0.015 and the learning rate was made as 0.0001.

4.4 Performance Comparison

Here, we present the comparative results of our proposed model with the current state-of-the-art methods of CF machine learning models. Our model focuses on the interrelationship between users and items.

- ***Average (AVG)****:* It ranks the items by taking the average of all ratings of items. This is usually used as the baseline for model evaluation.
- ***User-Item (U-I)****:* This algorithm has been used by Amazon for the user-item collaborative filtering method. It is one of the well-known methods for recommending items to related users [33].
- ***Fast Parallel Stochastic Gradient Descent (FPSGD)*** is a Matrix Factorization technique of optimizing the existing SGD method for faster execution performance [27].
- ***ALS*** is a well-known Matrix Factorization technique for recommendation system which uses a square loss function. We used the same parameters as reported in [34].
- ***Parallel Matrix Factorization (PMF)*** is an effective Matrix Factorization technique for a recommendation system [26].
- ***Distributive Matrix with Ensemble (DMWE)****:* This is our proposed model for recommender systems.

The proposed method (*DMWE*) has shown slightly better results on Movie Lens dataset, as compared to the *PMF*, *ALS* and *FPSGD* as shown in Table 2. In *FPSGD*, the authors focus on the parallelization of the work and faster execution of the algorithm, and not on the performance enhancements. Our method also outperformed the baseline and the state-of-the-art methods, including *User-Item* and *Average* as shown in Table 2. We arranged the basic CF model based on relative strength, where simple and less accurate models appeared on the left and arranged the more advanced/ highly accurate models on the right.

Table 2. RMSE of different Methods

Methods	ML4M
AVG	1.093
U-I	0.89
FPSGD	0.830
ALS	0.82989
PMF	0.82812
DMWE	**0.82501**

We also discuss the effect of the ensemble size on the integration of DM. We increase the base learners in our basic DMWE model up to 3. We adopt the same parameters for each member of the DM model and only use the Movie Lens 4M dataset to train the model. As we increase the number of base learners in the DMWE model, the accuracy results of our model are also improved as shown in Table 3.

Table 3. Size of the Ensemble

Ensemble size	RMSE
DMWE-2	0.82501
DMWE-3	0.82405

5. Conclusion

In this paper, we propose a novel Distributive Matrix Collaborative filtering method with an ensemble technique. Through the ensemble technique, we use users and items copy datasets for each component of the ensemble learner. In our proposed model, we make use of the explicit ratings in the input matrix and also use the same rating matrix for each learner of an ensemble. The experiments on benchmark dataset show the effectiveness of our proposed model. In future, we will attempt to apply the fusion concept for rating prediction in collaborative filtering to enhance the RMSE performance. We will validate our model with different and large-scale datasets of rating matrix to measure the efficiency of a recommender system. We will also use the explicit and implicit ratings in the collaborative filtering for recommender system.

REFERENCES

[1] Y. Koren and R. Bell, "Advances in collaborative filtering", in Recommender Systems Handbook, Second Edition, 2015.

[2] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, "Neural graph collaborative filtering", in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 165–174, 2019.

[3] S. Ghazarian and M. A. Nematbakhsh, "Enhancing memory-based

collaborative filtering for group recommender systems", Expert Syst. Appl., vol. 42, no. 7, pp. 3801-3812, 2015.

[4] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives", ACM Computing Surveys. vol. 52, no. 1, pp.38, 2019.

[5] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance", *Clim. Res.*, vol. 30, no. 1, pp.79-82, 2005.

[6] J. Bennett and S. Lanning, "The Netflix Prize", in *Proceedings of* KDD Cup and Workshop, 2007.

[7] Netflix, "Netflix Prize", *Proceedings of KDD Cup and Workshop*, 2009. [Online]. Available: www.netflixprize.com. [Accessed: 03-Jan-2019].

[8] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems", vol. 42, no. 8, pp. 30–37, 2009.

[9] R. Kumar, B. K. Verma, and S. Sunder Rastogi, "Social Popularity based SVD++ Recommender System", Int. J. Comput. Appl., vol. 87, no. 14, pp. 33-37, 2014.

[10] R. Burke, "Hybrid recommender systems: Survey and experiments", *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, 2002.

[11] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Scalable Collaborative Filtering Approaches for Large Recommender Systems", J. Mach. Learn. Res., vol. 10, pp. 623–656, 2009.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce", in Proceedings of the 2nd ACM conference on Electronic commerce - EC '00, pp. 158–167, 2000.

[13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms", in Proceedings of the 10th International conference on World Wide Web, pp. 285–295, 2001.

[14] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model", in Proceedings of the 14th ACM SIGKDD International conference on Knowledge Discovery and Data mining, pp. 426–434, 2008.

[15] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model," Knowl.-Based Syst., vol. 97, pp. 188–202, 2016.

[16] R. M. Bell and Y. Koren, "Scalable collaborative filtering with jointly derived neighborhood interpolation weights", in *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 43–52, 2007.

[17] Z. Zhang, Y. Liu and Z. Zhang, "Field-Aware Matrix Factorization for Recommender Systems," in *IEEE Access*, vol. 6, pp. 45690-45698, 2018.

[18] R. Bell and Y. Koren, "Lessons from the Netflix Prize Challenge", SIGKDD Explorations 9, 2007.

[19] L. Candillier, F. Meyer, F. Fessant. "Designing Specific Weighted Similarity Measures to Improve Collaborative Filtering Systems". IEEE International Conference on Data Mining, pp. 242–255, 2008.

[20] X. Su, T.M Khoshgoftaar, X. Zhu, and R. Greiner. "Imputation-boosted collaborative filtering using machine learning classifiers". Symposium on Applied Computing, Proceedings of the 2008 ACM symposium on Applied computing, pp.949–950, 2008.

[21] M. Wu "Collaborative Filtering via Ensembles of Matrix Factorizations". Proceedings of KDD Cup and Workshop, pp. 43–47, 2007.

[22] J.S Lee, J.-S., & S. Olafsson. "Two-way cooperative prediction for collaborative filtering recommendations". Expert Systems with Applications, vol. 36, no. 3, pp. 5353–5361, 2009.

[23] A. Schclar, A. Meisels, A. Gershman, L. Rokach, A. Tsikinovsky. "Ensemble Methods for Improving the Performance of Neighborhood-based Collaborative Filtering". Proceedings of ACM RecSys pp. 261–264, 2009.

[24] M. Jahrer, A. Töscher, R. Legenstein, "Combining predictions for accurate recommender systems", Proc. 16th ACM SIGKDD, pp. 693-702, 2010.

[25] Q. Ho, J. Cipar, H. Cui, and S. Lee, "More effective distributed ML via a stale synchronous parallel parameter server," in *Proceedings - Advances in Neural Information Processing Systems (NIPS)*, pp. 1223–1231, 2013.

[26] H. F. Yu, C. J. Hsieh, S. Si, and I. Dhillon, "Scalable coordinate descent approaches to parallel matrix factorization for recommender systems", in *Proceedings - IEEE International Conference on Data Mining (ICDM)*, pp. 765–774, 2012.

[27] Y. Zhuang, W.-S. Chin, Y.-C. Juan, and C.-J. Lin, "A fast parallel SGD for matrix factorization in shared memory systems", in *Proceedings of the ACM conference on Recommender systems*, pp. 249–256, 2013.

[28] H. Li, K. Li, J. An, and K. Li, "MSGD: A Novel Matrix Factorization Approach for Large-Scale Collaborative Filtering Recommender Systems on GPUs", *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 7, pp. 1530–1544, 2018.

[29] W. Tan, S. Chang, L. Fong, C. Li, Z. Wang, and L. Cao, "Matrix Factorization on GPUs with Memory Optimization and Approximate Computing", in *Proceedings of the International Conference on Parallel Processing*, 2018.

[30] X. Guan, C. Li and Y. Guan, "Matrix Factorization With Rating Completion: An Enhanced SVD Model for Collaborative Filtering Recommender Systems," in *IEEE Access*, vol. 5, pp. 27668-27678, 2017.

[31] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent", in *Proceedings of the 17th ACM SIGKDD International conference on Knowledge Discovery and Data mining - KDD '11*, pp. 69–77, 2011.

[32] F.Maxwell and J. A. Konstan, "The MovieLens Datasets: History and Context", *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 19, 2015.

[33] Y. Koren, "Factor in the neighbors: Factor in the Neighbors: Scalable and Accurate Collaborative Filtering", *ACM Trans. Knowl. Discov. Data*, vol. 4, no. 1, pp. 1–24, 2010.

[34] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the netflix prize", in *Proceedings of the International conference on Algorithmic Aspects in Information and Management*, pp. 337–348, 2008.