Model-Based Collaborative Filtering for Recommender Systems: An Empirical Survey

Abinash Pujahari
Department of Computer Science and Engineering
National Institute of Technology Raipur
Raipur, India
apujahari.phd2017.cse@nitrr.ac.in

Dilip Singh Sisodia
Department of Computer Science and Engineering
National Institute of Technology Raipur
Raipur, India
dssisodia.cs@nitrr.ac.in

Abstract— **Saving time is one of the most important things. A Recommender system certainly saves our time by generating a prediction of things for us from an extensive database that we might like. Due to the widespread use of the internet, it became a popular area of research in the machine learning community. It is being used in almost all web sites these days to keep track, what users want or like so that when they visit next time, it can predict something for them. A lot of research has been done on this field, but still, the performance of these systems is not that good to be accepted by the end-users. Among these, model-based collaborative filtering is widely used in commercial recommender systems due to its scalability and capability of handling sparse datasets. In this paper, we will provide a detailed insight into these techniques along with their advantages as well as disadvantages.**

Keywords—Recommender System, Collaborative Filtering, Model-Based

I. INTRODUCTION

Recommender Systems[1]–[3] are the tools or techniques that predict items, which might be useful for a user. An item can be products, songs, movies, etc. It also helps users in many ways like which products they should buy, which songs they may like etc. It is beneficial for those who could not find out the things from the vast database of items. Recommender Systems can be personalized or group-based. In a group-based recommender system, individual preferences may diverge from each other. But in the real world scenario, we often see personalized recommender systems, especially in the e-commerce sites. Personalized Recommender Systems (PRS) predicts things for an individual where the preferences of that individual are taken into consideration. The choices can be learned or collected from users' past experiences. Group Recommender Systems (GRS) [4], [5] are the systems where the recommendations are for a group of users and each member of the group has their preferences. GRS is complicated to maintain because we have to satisfy each member of the group.

Recommender systems generate recommendations based on two techniques. In the Collaborative Filtering technique, things or items are being predicted based on the similarity of the preferences among two users or a group of users. For example, user A liked three items (namely A, B, and C), and user B liked two out of them (namely A and B) then the system predicts item C as the recommended items to the user B, considering that preferences of both the users are same. Another way of collaborative filtering is that, if the user liked some items in the past, then the items having the same features as the items liked by the user in the past will be recommended. This is called item-item collaborative filtering techniques. Collaborative filtering again can be of three kinds, as shown in Figure 1. In neighbourhood-based filtering, it compares either similarity among users or similarity among items. In model-based, the system uses latent factors to predict the preference score of unseen items. Based on the ratings, the RS generates recommendations. The third one combines both memory-based and model-based.

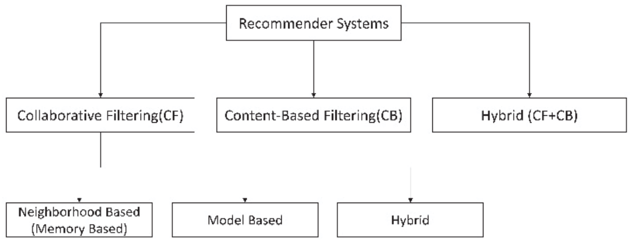

Figure 1 Techniques of Recommender Systems

Another way of making recommendations is the content-based filtering [6], [7] technique, where the system builds a model according to user preferences. In other words, it learns the users' preferences using machine learning techniques and then predicts things based on that learned model. It generally takes ratings as the users' input and determines the satisfaction of a product or item. But to apply this technique, the past behaviour of users' is needed. The performance of the recommender system relies on the learned model. The third category of recommender system is of the hybrid type, which combines the feature of both, i.e., Collaborative and Content-based filtering techniques, to generate recommendations. More sophisticated hybrid recommender systems have been proposed in the field of the recommender system.

This paper is organized as follows. Section II describes the recent researches done in model-based collaborative filtering for recommender systems. Section III critically analyses the model-based approaches in the field of recommender systems and their advantages as well as disadvantages. This section gives a clear view of the recent findings and methods in recommender systems. Section IV contains some experimental comparisons of the latest techniques. The conclusion includes a detailed summary of the survey as well as the critical shortcomings of the current research in the field of recommender systems.

II. RELATED WORKS

We will discuss the model-based CF techniques in two phases. In the first part, we will discuss factorization/ probabilistic models, and in the second phase, we will discuss the neural network-based models including deep learning-based models.

A. Factorization based Models

In this approach, the user-item rating matrix is considered as a dot product of user latent factors and item latent factors having the same dimension. The most preferred and widely used technique in this category is the Matrix Factorization [8]–[10]. This technique is capable of handling sparse data (i.e., when we have very few user ratings). Generally, in an RS, we have very few ratings available. To predict the unknown ratings for any user, we take the dot product of the corresponding user latent factor vector and item latent factor vector. Let R be the rating matrix, and U and V be the user and item latent feature matrices, respectively. Then the predicted rating for an item v for user u can be given as:

$$\hat{R}_{(u,v)} = U_u V_v \qquad (1)$$

Equation 1 can be used to obtain the predicted rating for any user towards any item. The user & item latent factors can be obtained through an optimization procedure using regularization terms as mentioned in Equation 2.

$$\min_{U,V} \sum_{(u,i)\in K} \left(R_{u,i} - \hat{R}_{u,i}\right)^2 + \lambda_U ||U||_F^2 + \lambda_V ||V||_F^2 \qquad (2)$$

The minimization of the objective function defined in Equation 2 is based on minimizing the sum of squared errors in the predicted ratings as compared to the actual ratings. Several enhancements have been made to factorization-based techniques for recommender systems. Alejandro et al. proposed an approach to recommender system based on Longest Common Subsequence (LCS)[11], where they have taken rating information of users as strings. The standard LCS algorithm is used in finding subsequence in two strings. The idea behind this approach is that, it can find out the common ratings between two users which represents similar items and similar users also. This method uses a threshold value which is the maximum difference between two ratings under which they can be treated as similar. Similarly, in case finding similarity among users it will calculate the maximum number of similar kind of ratings given by both the users. Consider two users x and y rated 5 items having rating vectors < 4,*,5,3,1> and <4,5,*,4,4> where '*' means missing rating value and let it be stored in 'sim'. They proposed a procedure $LCS\_CF(x, y, f, \delta)$, which returns the matched similarity between user x and y based on ratings vector written above. Then the similarity index between two users are proposed by Alejandro et al. is given in Equation 3. This similarity index is used in generating recommendation.

$$s(x, y) = \frac{(sim(x, y))^2}{|f(x)|.|f(y)|} \qquad (3)$$

Antonio Hernando et al. proposed a novel approach[12] to recommender system by using non negative matrix factorization with collaborative filtering approach. They also proposed a Bayesian probabilistic model for the collaborative filtering technique. This paper uses user ratings as non-negative matrices which factorizes the value in between [0,1]. This paper also outlines the disadvantages of classical matrix factorization techniques, because of its inconsistencies in the predicted rating values for the respective users. The probabilistic model generates the prediction based on the rating matrix generated earlier. One example of matrix representation is given in Table I, where '*' represents no rating information is present. After this they used some normalization technique to fill out the missing values.

Table I USER-ITEM RATING MATRIX

Users	Items			
	I_1	I_1	I_1	I_1
U_1	5	5	3	5
U_2	5	*	4	*
U_3	*	5	*	1
U_4	*	3	2	*
U_5	4	*	1	*

After doing the matrix factorization, they used the probabilistic technique to find out whether an user liked an item or not by using the formula given in Equation 4. The value of $a_{uk}.b_{ki}$ can be calculated from the matrix factorization table generated.

$$p_{ui} = \sum_{k=1}^{K} a_{uk}.b_{ki} \qquad (4)$$

J. Bobadilla et al. proposed a new collaborative filtering technique[13] which uses the common and in common votes between users for the items or things. This paper basically uses user votes as ratings and calculates the intersection of items for a set of users by using methods like Jaccard, Mean Squared differences etc. Before generating recommendation this paper first generate set of k-neighbor for each user, who have similar kind of preference to the original user. This paper calculates the MSD (Mean Squared Difference)[14] between two users, where two users are mostly similar if the MSD value tends towards zero and two users are different if the value tends towards 1. Consider the normalized rating value between two users $(x, y), rx\{0.65, 0.45, 0.78, *, 0.86\}$ and $ry\{0.68, 0.54, *, 0.77, *\}$. The distance between users x and y for each item 'i' by the formula given in Equation 5.

$$dist_{x,y}^{i} = (r_x^i - r_y^i)^2 \forall\, i \mid r_x^i \neq * \vee r_y^i \neq * \qquad (5)$$

After calculating the distance, the MSD value is calculated using the arithmetic average of the values present in the 'dist' vector. After calculating the distance measure between two users, a similarity measure based on Jaccard similarity is used in this paper which is represented in Equation 6. This similarity is used while predicting any item for the respective users. These similarity measures are integrated while generating recommendation for any user using collaborative filtering technique.

$$J(x, y) = \frac{r_x \cap r_y}{r_x \cup r_y} = \frac{|dist_{x,y}|}{|r_x| + |r_y| - |dist_{x,y}|} \qquad (6)$$

B. Neural Network Based Models

Neural network-based models are popular in machine learning community and it is no exception in the field of recommender system. Several neural network models have been proposed towards recommender system. One of the significant approaches to neural network-based model was proposed by Salakhutdinov et al. [15] in which they have used Restricted Boltzmann machine for recommender system. Later, an enhancement to this model was proposed by Pujahari and Sisodia [16] by implementing RBM in a PR-based framework.

Among other models, CDAE[17] was one of the promising approaches which uses denoising auto encoder in a collaborative filtering framework. Similar works were done after then using CDAE in multi-layer neural network framework. Neural network-based factorization machine also comes to existence[18], which uses multi-layer neural framework to factorize user-item latent feature matrices.

III. EVALUATION MEASURES

Various evaluation measures have been proposed in the field of recommender system to find out the efficiency of prediction of any recommender system. There is no standardized formula or evaluation measure available to evaluate the performance of any recommender system. This section describes some of the benchmark evaluation measures used in the recent research to calculate the prediction performance of recommender system.

A. Measures for Prediction Accuracy

These measures are used to test the predicted ratings or find the error in predicted ratings. The most popular among these measures is RMSE (Root Mean Squared Error). It is the most widely used measure to find out the accuracy of predicted ratings of any recommender system. Because, if the accuracy of the predicted rating is accurate then the predicted items also will be accurate for any user. In many recommender system ratings we have the test data set where we compare the learned ratings with the actual ratings. Suppose we have a test data set 'T' where the user-item is taken as pair (u, i) in which user ratings '$r_{u,i}$' are known. Then, $\hat{r}_{u,i}$ are the predicted ratings. The RMSE [2] between actual ratings and the predicted ratings is given in Equation 7.

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{u,i \in T} (\hat{r}_{u,i} - r_{u,i})^2} \tag{7}$$

Mean Absolute Error (MAE) is another measure, based on RMSE, which is also frequently used in finding accuracy of predicted ratings. The MAE value can be calculated using Equation 8.

$$MAE = \frac{1}{|T|} \sum_{u,i \in T} |\hat{r}_{u,i} - r_{u,i}| \tag{8}$$

B. Measuring Usage Prediction

Evaluating the usage prediction is based on the confusion matrix which is used in any data mining algorithm evaluation. The modified equations of finding accuracy, precision and recall with respect to recommender system is given in Equations 9, 10, and 11 respectively. Where R is the set of recommended items which is generated by the RS. U is the set of items used by a user. Here, $\bar{R}$ is the list of items, that are not recommended from to total number of items. We can calculate the accuracy, precision and recall values for the recommender system by taking the average value for all the users in the system.

$$Accuracy = \frac{|(R \cap U)| + |\bar{R} \cap \bar{U}|}{|U|} \tag{9}$$

$$Precision = \frac{|R \cap U|}{|R|} \tag{10}$$

$$Recall = \frac{|R \cap U|}{|R \cap U| + |R \cap \bar{U}|} \tag{11}$$

C. Ranking Based Measures

Ranking based measures are used to measure the ordering of items generated by a recommender system. One of the standard ranking measure used for evaluating ranking is NDCG@K (Normalized Discounted Cumulative Gain)[19], where 'K' defines the position. The formula for the same is given in Equation 12 and 13. Another measure most frequently used is 'Precision@K' which is just an extension to precision measure as discussed earlier but is calculated with respect to position of the recommendation list.

$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{12}$$

$$DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{13}$$

IV. EXPERIMENTAL ANALYSIS

Before we can do the comparison among different model based collaborative filtering technique, we will discuss different dataset available for recommender system testing.

Standardized and consistent data sets was a major issue in the field of recommender system research earlier. But, thanks to the era of internet, some benchmark data sets are available in the internet which promotes the research in this domain. The Movie Lens data sets [20] is one of them. Different volumes of the same kind of data is available at MovieLens. Each data set contains movie information having different genres. Besides this there is another file which contains the rating information of different users given to those movies. This data set can be used to analyze or learn to build a personalized as well as group recommender system. Most of the papers related to recommender system uses this data set to evaluate their model. Jester data set [21] is another one which contains jokes from different region and is basically joke rating system. The ratings of the jokes are given in terms of how funny it is. The rating values are between -10 to 10.

Another popular data set available in the internet is 'Last.fm'[22] which is an online music recommender system. This data set contains the most listened songs or artists for all the users including the number of times being played by those users. The data set also uses a separate file which contains the users' tags for the songs, which can be used to build a vector. A comparison among different data sets in the field of recommender system is given in Table II. The stats given in this table has been obtained from Kdnuggets[1]. The post is originally posted by Alexander Gude.

[1] https://www.kdnuggets.com/2016/02/nine-datasets-investigating-recommender-systems.html

Table II DESCRIPTION OF DATASETS AVAILABLE FOR RECOMMENDER SYSTEMS

Dataset	Users	Items	Ratings	Density	Rating Scale
MovieLens 1M	6040	3883	1000209	4.26	[1, 5]
MovieLens 10M	69878	10681	10000054	1.33	[0.5, 5]
MovieLens 20M	138493	27278	2,00,00,263	0.52	[0.5, 5]
Jester	124113	150	5865235	31.50	[-10, 10]
Book-Crossing	92107	271379	1031175	0.0041	[1, 10]
Last.fm	1892	17632	92834	0.28	Play counts
Wikipedia	5583724	4936761	417996366	0.0015	Interactions
Openstreetmap	231	108330	205774	0.82	Interactions
GIT	790	1757	13165	0.95	Interactions

A. Comparison of Factorization based Models

In this section, we will provide some key comparison among factorization based models. We have used the MovieLens-1M dataset for all the experimental comparison. All the parameter setting for the RS models used in this comparison is obtained from respective article. But the same training and testing dataset is used for all the RS models for unbiased comparison. We have used 70% of the total ratings from ML-1M as training and remaining for testing. In the factorization based model, the user-item latent feature matrix dimension plays an important role towards the performance of the models.

The performance of different models using NDCG measure with respect to dimension(D) of user item latent feature matrices is given in Table III. The values obtained for each method is based on a Top-10 recommendation task. Here, we can observe that, the performance is increased when we increase the dimension and reaches a saturation point when the performance cannot the improved further.

Table III PERFORMANCE COMPARISON WITH RESPECT TO DIMENSION

Models	Dimension					
	D=10	D=20	D=30	D=40	D=50	D=60
SVD	0.3215	0.3456	0.3891	0.4073	0.4128	0.4154
SVD++	0.3376	0.3501	0.4056	0.4311	0.4415	0.4441
PMF	0.3365	0.3614	0.4107	0.4489	0.4510	0.4532
MCRI	0.3765	0.3924	0.4321	0.4675	0.4736	0.4755
PMF+P+N	0.3843	0.4040	0.4373	0.4521	0.4613	0.4624
LCS	0.3921	0.4012	0.4263	0.4456	0.4639	0.4670
PDP-PMF	0.4329	0.4417	0.4735	0.4920	0.5016	0.5041
DVMF	0.4353	0.4476	0.4821	0.4932	0.5124	0.5144

Next, we have studied the training time with respect to dimension of each model. Figure 2 shows the training time in seconds per user with respect to dimension for different model. Here also we can see that, the training time increases when the dimension increases which is obvious. The training time per user for SVD and PMF is less since they only consider the user item latent feature matrices which needs to be updated in each iteration. In our case the saturation point of dimension is 50. However, methods like MCRI [23], PDP-PMF[24] and DVMF [25] uses other matrices (like user-item features, side information) which need to be updated along with user item latent factors in each iteration of the optimization process. Hence, longer time required for each iteration, as displayed in Figure 2.

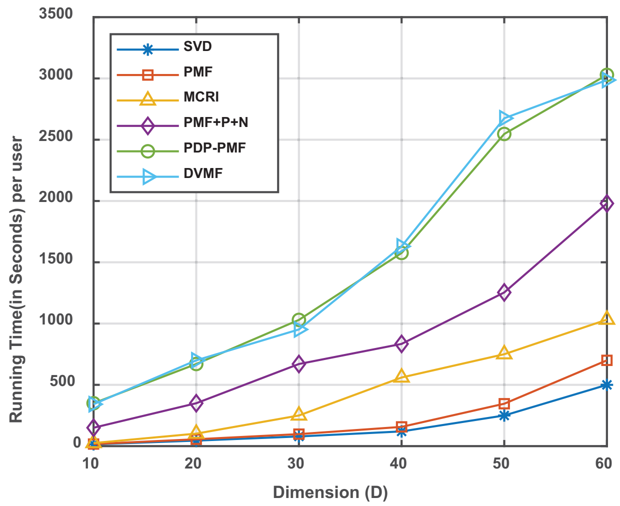

Figure 2 Training time per user with respect to dimension

B. Comparison of Neural Network Models

Here, we will provide the comparison among different neural network based models. Among neural network based we have chosen the models: CDAE[17], NeuMF[26], NFM[18] and NAIS[27]. The same training and testing set, as discussed earlier, has been used for comparing all these models. For comparison we will use NDCG measure at different positions. Table IV shows the results obtained using the above mentioned methods at different positions using NDCG measures. It can be seen that, NAIS is performing better because of integration of user-item feature in a neural network framework.

Although the performance of neural network based models are good in terms of ranking, their running time is much more than the traditional factorization based models, as observed during experiment.

Table IV COMPARISON AMONG NEURAL NETWORK MODELS

Models	NDCG@K				
	K=5	K=10	K=15	K=20	K=30
CDAE	0.4129	0.4365	0.4034	0.3872	0.3578
NeuMF	0.4567	0.4738	0.4241	0.4136	0.3975
NFM-1	0.4602	0.4768	0.4327	0.4148	0.4011
NAIS	0.4639	0.4935	0.4563	0.4387	0.4233

V. CONCLUSION

This paper analyzes the various model-based collaborative filtering techniques available for recommender systems. We have also compared the different factorization and neural network-based models. In case of factorization-based model the dimension of the user item latent factor matrices plays a crucial role towards the performance system as well as the computational complexity of the recommender system. Also, the user-item features (i.e. side information) is necessary to obtain exact preference when the dataset is too sparse. The model that incorporate this information is highly desirable. In our future work we will exploit this information to build an efficient recommendation model, that can generate better predictions even when the dataset is too sparse.

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” Knowledge-Based Syst., vol. 46, pp. 109–132, Jul. 2013.

[2] P. Resnick and H. R. Varian, “Recommender systems,” Commun. ACM, vol. 40, no. 3, pp. 56–58, 1997.

[3] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, 2005.

[4] A. Pujahari, V. Padmanabhan, and S. Patel, “Nearest neighbour with priority based recommendation approach to group recommender system,” in Advances in Intelligent Systems and Computing, 2016, vol. 411, pp. 347–354.

[5] W. Wang, G. Zhang, and J. Lu, “Member contribution-based group recommender system,” Decis. Support Syst., vol. 87, pp. 80–93, 2016.

[6] P. Lops, M. De Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in Recommender systems handbook, Springer, 2011, pp. 73–105.

[7] A. Pujahari and V. Padmanabhan, “An approach to content based recommender systems using decision list based classification with k-DNF rule set,” in Proceedings - 2014 13th International Conference on Information Technology, ICIT 2014, 2014, pp. 260–263.

[8] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer (Long. Beach. Calif)., no. 8, pp. 30–37, 2009.

[9] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” Nature, vol. 401, no. 6755, p. 788, 1999.

[10] Y. Wang and Y. Zhang, “Nonnegative Matrix Factorization: A Comprehensive Review,” IEEE Trans. Knowl. Data Eng., vol. 25, no. 6, pp. 1336–1353, Jun. 2013.

[11] A. Bellogín and P. Sánchez, “Collaborative filtering based on subsequence matching: A new approach,” Inf. Sci. (Ny)., vol. 418–419, pp. 432–446, 2017.

[12] A. Hernando, J. Bobadilla, and F. Ortega, “A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model,” Knowledge-Based Syst., vol. 97, pp. 188–202, 2016.

[13] J. Bobadilla, F. Serradilla, and J. Bernal, “A new collaborative filtering metric that improves the behavior of recommender systems,” Knowledge-Based Syst., vol. 23, no. 6, pp. 520–528, 2010.

[14] J. Bobadilla, A. Gutiérreza, F. Ortegab, and B. Zhuc, “Reliability quality measures for recommender systems,” Inf. Sci. (Ny)., vol. 442–443, pp. 145–157, 2018.

[15] R. Salakhutdinov, A. Mnih, and G. Hinton, “Restricted Boltzmann Machines for Collaborative Filtering,” in Proceedings of the 24th international conference on Machine learning, 2007, pp. 791–798.

[16] A. Pujahari and D. S. Sisodia, “Modeling Side Information in Preference Relation based Restricted Boltzmann Machine for recommender systems,” Inf. Sci. (Ny)., vol. 490, pp. 126–145, Jul. 2019.

[17] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, “Collaborative Denoising Auto-Encoders for Top-N Recommender Systems,” in Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, 2016, pp. 153–162.

[18] X. He and T.-S. Chua, “Neural Factorization Machines for Sparse Predictive Analytics,” in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 355–364.

[19] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” ACM Trans. Inf. Syst., vol. 20, no. 4, pp. 422–446, 2002.

[20] F. M. Harper and J. A. Konstan, “The MovieLens Datasets,” ACM Trans. Interact. Intell. Syst., vol. 5, no. 4, pp. 1–19, 2015.

[21] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, “Eigentaste: A constant time collaborative filtering algorithm,” Inf. Retr. Boston., vol. 4, no. 2, pp. 133–151, 2001.

[22] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, “The Million Song Dataset,” in Proceedings of the 12th International Conference on Music Information Retrieval ({ISMIR} 2011), 2011.

[23] Y. Fang and L. Si, “Matrix co-factorization for recommendation with rich side information and implicit feedback,” in Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec), 2011, pp. 65–69.

[24] S. Zhang, L. Liu, Z. Chen, and H. Zhong, “Probabilistic matrix factorization with personalized differential privacy,” Knowledge-Based Syst., vol. 183, p. 104864, 2019.

[25] W. Zhang, X. Zhang, H. Wang, and D. Chen, “A deep variational matrix factorization method for recommendation on large scale sparse dataset,” Neurocomputing, vol. 334, pp. 206–218, 2019.

[26] 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, 2017, pp. 173–182.

[27] X. He, Z. He, J. Song, Z. Liu, Y. Jiang, and T. Chua, “NAIS: Neural Attentive Item Similarity Model for Recommendation,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 12, pp. 2354–2366, Dec. 2018.