Vector Space Model Embedding for Recomender System Neural Networks

Yahui Wang, Paul Craig
Department of Computer Science and Software Engineering,
Xi'an Jiaotong-Liverpool University, Suzhou, China,
Yahui.Wang16@student.xjtlu.edu.cn,
P.Craig@xjtlu.edu.cn

***Abstract* — Research shows that recommendation algorithms such as Collaborative Filtering (CF) can be enhanced using neural network (NN) to make more accurate recommendations. This project proposes an adaptive recommendation model based on NN to quickly access different items. A vector space embedding method is used to vectorize users and items before a Deep Neural Network (DNN) rating prediction network model is used to predict users' rating behavior. Multi-domain datasets are utilized in experiments to evaluate results and compared our method with traditional recommendation algorithms. Results show that the improved NN based recommendation model is more effective and achieves a higher score in both similarity calculation and predication.**

Keywords— Recommendation Algorithms; Versatility; Neural Network; Embedding Method; Rating Prediction Network

I. INTRODUCTION

Due to developments such as the rapid development of social networks, which increase the quantity of potentially valuable data generated by individuals, individuals are finding it increasingly difficult to find information that is actually valuable to them. Recommendation algorithms are now widely be used in various fields [1] as the principle method for solving this problem.

In the aspect of Personalized recommendation algorithms, the most widely used is collaborative filtering (CF) [2]. A large amount of websites based on the collaborative filtering have proposed improved models and achieved significant results [3]. With the increased use of neural networks in various fields, the traditional CF algorithm has also been combined with neural networks [4]. In this way, it could make more accurate recommend results based on user features and behavioral data learning [5].

However, there are many different kinds of products in various enterprises. Some user service products integrate the demand of recommendation function. What's more, the application scenario of lots of products are very similar. If we implement a specific recommendation system for each product, the cost will be huge in terms of development and maintenance. Because of the automatic extraction of abstract features in neural network computation, it is possible to construct a adaptive and general recommendation algorithm based on neural networks [6].

There are some changes to enable the cross-domain versatility for recommendation models based on neural networks. Firstly, Neural networks are not sensitive to sparse data. Most rating or behavior vectors of users are sparse and expression of the data becomes a problem. Secondly, the efficiency of candidate set calculation in neural networks is low. It is a change to reduce the cost in computing. Thirdly, it is necessary to propose a reasonable neural network model for accurately predicting a user's rating. The training for this model faces the problem of unstable gradient and over fitting [7].

In order to solve these problems, this paper makes the following contributions:

- We provide an adaptive and general recommendation model for business to quickly access different products.
- We use Vector Space Model Embedding to generate a lower dimensional continuous feature vector in order to avoid the problems associated with sparce data.
- We calculate the candidate set by computing the similarity between vectors.
- We propose the rating prediction network model to optimize the neural network.

The rest of this paper is constructed as follows. Section II introduces the related work. Section III illustrates our methodology and model for recommendation. Section IV gives the evaluation models and the experiment results. Finally, the conclusion and further work are provided in section V.

II. RELATED WORK

Related research in the area of recommendation methods can be categorized into three areas as follows:

The first model is based on the Collaborative Filtering (CF) Algorithm. In 2003, Linden published the model of the CF algorithm based on items used by Amazon [8]. Experimental results show that CF algorithm based on items has a higher accuracy than the CF algorithm based on users.

The second strategy is based on Matrix Factorization. Koren [9] and Salakhutdinov [10] analyze how matrix decomposition can be used to reduce the dimensionality of an items rating matrix to efficiently process large data sets. Gao et al.[11] used a kind of language-based approach for location-based social networks (LBSNs) to provide location recommendations. Shin and others[12] use Word2Vec to build feature-based word models to recommend blogs for users.

The third method is neural network models in personalized recommendation. Among them, the early representative application of the neural network model is that of Ruslan et al. [13] who constructed a Restricted Boltzmann Machine (RBM) to realize CF in 2007. RBM is a kind of generative random neural network. The RBM is used to model user behavior and filter the data accordingly.

In 2013, C.J.C. Burges et al. [14] analyzed the content of music using a deep learning method, and used the content feature set to improve the accuracy of a recommendation system. This greatly improved the accuracy of the algorithm. With the service of Spotify [15], deep learning is combined with the traditional CF. In 2015, Microsoft research [16] applied neural networks in the field of news and application recommendations.

There are some other recommendation methods that could be used as the basis of further study. Because traditional CF recommendation algorithms have many defects, especially if user behavior data is not sufficient, recommender accuracy is generally far from ideal. Although a matrix factorization technique can ease the data sparseness problem to some extent, it needs to compute an additional user and item feature matrix. Neural networks have the ability to automatically extract abstract features. Not only according to user's behavior in the process of training data, also can join the features of users and items help network data fitting. This should allow us to predict user behavior more accurately.

III. Model FOR Recommendation

This project proposes an adaptive recommendation model based on NN to quickly access different items. A vector space embedding method is used to vectorize users and items before a Deep Neural Network (DNN) rating prediction network model is used to predict users' rating behavior.

A. Framework

Firstly, the recommendation model utilizes the Vector Space Model Embedding method to vectorize the users and items and calculate the item candidate sets. Next, a neural network model is used to train and study the existing data in the way of vectorization. Thus the rating prediction network model can be obtained. After the model training is completed, the candidate sets are extracted as the input of the rating prediction network. Finally, the items in the candidate sets are ranked, so that the top items are obtained. The framework of this recommendation model is shown in Fig.1.

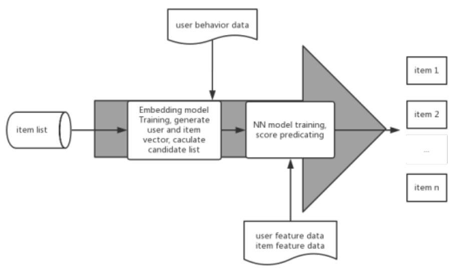
user behavior data
item list
Embedding model Training, generate user and item vector, caculate candidate list
NN model training, score predicating
user feature data
item feature data
item 1
item 2
...
item n

Fig.1. Framework for Recommendation Method based on NN

The model application process can be divided into 5 steps. It is shown as following:

Step 1: According to the collected data preprocessing and feature analysis, this project could get the overall characteristics of the data sets to help define the network structure. Meanwhile, the user behavior data is processed to obtain the user behavior context and item context.

Step 2: The generated context will be trained by Embedding method, with users and items. Word2Vec [17] will be used in Embedding method training.

Step 3: The candidate set of recommended items is calculated according to the similarity between vectors.

Step 4: Using the user vector and item vector generated in step 2, combining the user characteristics and item characteristic data, as the input of the NN model, to train the rating prediction neural network. Then, network parameters will be randomly initialized. The network model would be optimized by using the back propagation algorithm[18], Adam algorithm to speed up the training, and combining with Batch Normalize (BN) [19] and Dropout regularization method [20]. The expected output is user's behavior.

Step 5: The input is item list of candidates. Through the trained NN model, predicting user ratings to the candidate set items. The top items after sorting will be returned as the recommendation results.

B. Algorithm Specification

This section provides the regulations and instructions for the input data accepted by the recommendation model based on NN.

1) Data Specification

Behavioral Data: NN model is designed to predict user behavior, then it can provide support for recommendation. Therefore, user behavior act as labels in the training of the whole model to process supervised learning.

Feature Data: The user feature data and item feature data can represent users and items. Feature data is frequently used in content based recommendations and in knowledge reasoning based recommendations. These data can help NN fitting. Feature data can be divided into continuous data and discrete data. Continuous data: data such as user age, online time, etc. If the information such as day, month, year, is contained in time feature data, on the basis of the first time in the feature space, set it to 0. Then, the time will be transformed with the smallest granularity as measurement units in time feature data. This can ensure the data consistency. discrete data: such as binary or multivariate classification data, user gender, label data of item characteristics.

Overall, formula (1) shows the illustration of user feature and item feature.

$$\begin{aligned} &u_a : \{UCF_{1i}, UCF_{2j}, \ldots, UCF_{nk}, UDF_{1x}, UDF_{2y}, \ldots, UDF_{mz}\} \\ &i_b : \{ICF_{1i}, ICF_{2j}, \ldots, ICF_{nk}, IDF_{1x}, IDF_{2y}, \ldots, IDF_{mz}\} \end{aligned} \tag{1}$$

where n in UCF_{nk} means the n^{th} continuous feature of user a; k is the value in the feature space list; m in UDF_{mz} means the

m^{th} discrete feature; z is the value in the feature space list. Similarly, ICF is the continuous feature of items; IDF is the discrete feature of items. Each different feature has the independent feature space.

Data Expression: The network will use the user behavior data and the discrete feature and continuous feature to make the prediction. The neurons calculate and get the activate vlues of input by activating function, with continual iteration, transmission to obtain the output of the whole network. Therefore, the above data is not suitable for direct input to the network. It needs to be processed.

Step 1: User Vectorization and Item Vectorization

User behavior data is used to Embedding the items. This can reduce the dimension of users and items and retain the interactive information between them. This method can be used to calculate the similarity of users and objects, which can not only avoid the problem of data sparse in traditional algorithms, but also can be used as the input of the rating prediction NN model.

Step 2: Continuous Features Regularization

The continuous features can be scaled by using cumulative distribution $x = \int_{-\infty}^{x} df$. This can map the original x continuous feature obeying f distribution to $[0,1]$ space. The mapped x is the input of network. In this way, the influence of data size to network can be reduced.

Step 3: Discrete Features Vectorization

Discrete features such as classification of items, are often converted to numeric values when they enter the network. Each category corresponds to a separate number. The more categories of a particular feature, the greater the number space. The sensitivity of the network to large values makes this a problem for network computing. So this project will vector it. Features and features, categories and the categories are independent existence. In this case, the discrete features can be vectored directly.

Calculate Candidate Set: This step takes advantage of the Embedding method to generate candidate set, calculating the nearest K number of users' similarity and the nearest N number of items' similarity. The cosine distance is the relative position here. The weighted formula of the user's score for items and relative position is as follows:

$$score_i = \sum_{j=1}^{K} \frac{1}{distance_j} * rate_j \tag{2}$$

where $distance_j$ is the distance from the j^{th} user to the current target user among the k nearest users; $rate_j$ is the rating of user j for current item.

2) Rating Prediction Network Structure

The features of the previous section are the input of this network. The output is the prediction of rating. The input is composed of four parts: first is the item vector and user vector, then is continuous features with regularization, the last one is the discrete features. Discrete features input into the Embedding layer first, then the output of Embedding layer connects to the network. They are vector-trained by combining with the entire network. The long vector can be obtained after the input is spliced. s the structure of input layer in the rating prediction network.

The whole vector input into three hidden layers. The hidden layers use ReLU as the activation function. ReLU is a linear neural. The form is shown in the following formula:

$$f(x) = \max(0, x) \tag{3}$$

The gradient instability can be suppressed by using ReLU as the activation function. because of the nature of the function itself, the network will converge faster than the traditional Sigmoid neuron.

The last step is the output layer of the network. It is a single neuron. The neuron activation function is ReLU5. The formula is shown as following:

$$f(x) = \min(\max(0, x), 5) \tag{4}$$

When the input of ReLU5 is between 0 and 5, the output value is equal to the input value; The output is 0 when the input is less than 0; The output is 5 when input is greater than 5. Here basically have rating behavior and like/dislike binary behavior. ReLU can meet the flexibility and and maps the results simply. After the reshape operation, the prediction results can be find.

Finally, the structure of the whole rating prediction network model is shown in Fig.2

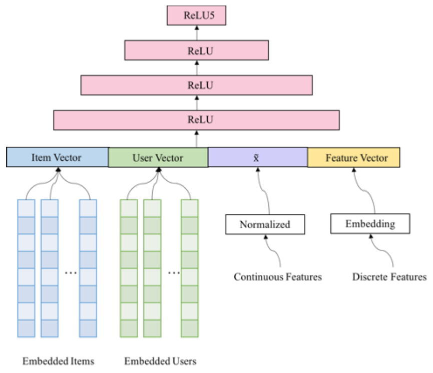

Fig.2. Structure for Rating Prediction Network Model

The illustration of neuron in rating prediction network is shown in table I. num represents the number of neurons in the input layer; each user and each item express with 1×50 dimensional vector; n_c is the number of continuous features, n_D on behalf of the number of discrete features; h_1 is the number of neurons in the first hidden layer; h_2 is the number of neurons in the second hidden layer; $h_3$is the number of neurons in the third hidden layer.

TABLE I. THE STURCTURE OF NEURON IN RATING PREDICTION NETWORK

	Numbers of Neurons
Input layer	$\text{num} = 1\times 50 + 1\times 50 + n_C + n_D \times 10$
The first hidden layer	$h_1 = \begin{cases} 1.5\times num, & 1.5\times num \le 512 \\ 512, & 1.5\times num > 512 \end{cases}$
The second hidden layer	$h_2 = \frac{1}{2}\times h_1$
The third hidden layer	$h_3 = \frac{1}{2}\times h_2$
Output layer	1

C. *Network Optimization*

there are a large number of parameters in network training. The model is more likely to happen over fitting problem. If only using back propagation and the method based on gradient, the network will converge slowly. What's more, it's easy to occur over fitting.

1) *Training optimization algorithms*

The optimization algorithm can be used to help the model automatically adjust parameters. The most widely used is Stochastic Gradient Descent (SGD). However, it relies on a fixed learning rate. When the rate is small, the network convergence will be slow. If it is larger, the minimum point may be missed or up and down near the extremum.

Adaptive Moment Estimation (Adam) is a kind of optimization algorithm using gradient. The characteristic is adaptive learning. Adam is more suitable for big data sets and higher-dimensional space.

2) *Regularization*

The most commonly used methods for resolving over fitting are generally referred to as regularization methods [21]. In addition to L1 and L2 regularization, this project adopts Dropout mechanism to help suppress over fitting in the training process.

3) *Inhibit gradient instability*

While ReLU replaced Sigmoid[22] as activation function can inhibit the part of the case of gradient disappear, but it is not completely solved and it may be a gradient explosion. Therefore, BN is used to help inhibit unstable gradient. The acceleration of BN for network training reliefs the Dropout effects on training speed to some extent.

IV. EVALUATION MODELS AND EXPERIMENT RESULTS ANALYSIS

This section introduces the datasets and data characteristics, and then conducts the model for evaluating the proposed recommendation algorithm. Finally, the performance results are compared between each dataset with the controlling variable method.

A. *Datasets*

this paper adopts MovieLens1M[23], Epinions665K[24], CiaoDVD[25] three open source datasets to verify the versatility of the algorithm. MovieLens is a typical recommendation website on there users rate movies from 1 to 5. Epinions is a consumer review community. Consumers can read reviews of various items to decide purchase. What's more, consumers can select trust you or distrust you based on the reviews. Ciao offers the reviews, consumer advice and product ratings by users on DVDs. Experiment setting of the datasets is illustrated in Table II.

TABLE II. SATISTICS OF DATASETS

Data sets	Users	Items	Rating Numbers	Rating Scope	Rating Density
MovieLens1M	6,040	3,706	1,000,209	[1,5]	4.47%
Epinions665K	40,163	139,738	664,824	[1,5]	0.0118%
CiaoDVD	7,375	99,746	278,483	[1,5]	0.0379%

In order to ensure that each user can be trained, users are the units for datasets districts. for each user, if the rating information is more than 10, then 20% rating data is selected as the test set. In other cases, select a score into the test set.

B. *Model for Evaluation*

In this papeer, the dimension of Embedding is defined as 50 dimensions according to the size of data set and the sparse degree of data. In terms of the distance selection in similarity calculation, this project uses Euclidean distance[26] to measure the distance.

Confusion matrix[27] allows visualization of the performance of this algorithm. It can be used in this experiment to decide which recommendation has the better performance. It is shown in Table III.

TABLE III. CONFUSION MATRIX

	results of experiments (Classified)	
True Classification	True Positive (TP)	False Negative (FN)
	False Positive (FP)	True Negative (TN)

Precision (5), Recall (6) methods are utilized to evaluate the performance based on the confusion matrix.

$$\text{Precision} = \frac{TP}{(TP+FP)} \tag{5}$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \tag{6}$$

NN training step utilizes the root-mean-square error (RMSE) as the cost function. The formula is show as following:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(f_i - y_i)^2}{n}} \tag{7}$$

Where f_i is the predicated value, y_i is the actual value. Mean absolute error (MAE) and RMSE evaluation indicators are used in terms of the comparison between test sets results. The formula of MAE is shown as following:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \qquad (8)$$

MAE can calculate the mean for the absolute error value of the whole space $|e_i| = |f_i - y_i|$.

Because the square-root error contains the square root, the final result could be very sensitive if the predicted value and the actual value have the relatively larger deviation. It is obvious that the smaller of value of MAE and RMSE, the predicted value is closer to the real value.

C. *Experimental Results*

The controlling variable method is used to analyze results of experiments.

The experiment results of MovieLens-1M and Epinions655K datasets are compared with Embedding method. Because of the rating information is sparse in Epinions655K, the expected result is that The lower performance with the mentioned recommendation methods in Epinions655K than MovieLens-1M. However, compared with the traditional filter CF algorithm, the performance is improved based on the Embedding method. Modeling for the current users or items with the form of context can certainly ease the effects of sparse data. The results is shown in table IV.

TABLE IV. THE CACULATION OF CANDIDATE SET USING EMBEDDING METHOD

Dataset	MovieLens1M			Epinios655K		
	Precision	Recall@3	Recall@5	Precision	Recall@3	Recall@5
ItemEm-CBOW	48.16%	29.41%	31.47%	1.78%	3.42%	3.88%
ItemEm-SGNS	47.27%	29.38%	31.38%	1.80%	3.35%	3.90%
UserEm-CBOW	43.14%	27.13%	29.31%	1.65%	3.13%	2.79%
UserEm-SGNS	44.62%	26.98%	29.73%	1.58%	2.98%	3.19%
CF-User	42.18%	25.75%	27.68%	0.63%	2.67%	2.11%
SVD	46.51%	28.90%	30.88%	0.68%	1.17%	1.76%

Finally, in order to verify the performance of the proposed rating predication network, this project conducts experiments on the three open source datasets and compare the experimental results with SVD++[28], BiasedMF[29], Restricted Boltzmann Machine (RBM) algorithms[30]. The experiments repeat several times to eliminate the influence of the initial factor.

TABLE V. THE COMPARESION OF RATING PREDICTION NETWORK RESULTS

	MovieLens1M	Epinions665K	CiaoDVD
UserAvg-MAE	0.829	0.928	0.781
ItemAvg-MAE	0.782	0.825	0.760
SVD++-MAE	0.667	0.818	0.752
BiasedMF-MAE	0.675	0.814	0.749
RBM-MAE	0.645	0.819	0.743
Rating Prediction Network-MAE	**0.581**	**0.804**	**0.717**
UserAvg-RMSE	1.035	1.199	1.031
ItemAvg- RMSE	0.979	1.094	1.026
SVD++- RMSE	0.851	1.057	1.001
BiasedMF- RMSE	0.853	1.048	0.976
RBM-RMSE	0.831	1.060	0.974
Rating Prediction Network- RMSE	**0.753**	**1.047**	**0.969**

From table V we can see the proposed model achieves better results among three datasets. The effect on Movielens datasets is more obvious than other two datasets. It is observed that neural network models on CiaoDVD and Epinions have some degree of over fitting. It shows that the training of neural networks depends on a large number of data resources. In addition, SVD++ algorithm and BiasedMF algorithm choose different hyper-parameter settings for different datasets. However, the hyper-parameter in the rating prediction neural network is appropriately scaled based on training data. Therefore, this neural network model has certain universality.

V. CONCLUSION AND FUTURE WORK

In this paper, an improved recommendation model based on neural networks with vector space model embedding method is introduced. Three open source datasets (MovieLens1M, Epinions665K, CiaoDVD) are used to evaluate the algorithm. The embedding method to generate a candidate set and the rating prediction network Model are evaluated seperately. The results of experiments indicate that the improved model is more effective than the original SVD++, BiasedMF and RBM methods.

REFERENCES

[1] B. Xiao and I. Benbasat, "E-commerce product recommendation agents: use, characteristics, and impact," *MIS quarterly,* vol. 31, pp. 137-209, 2007.

[2] 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*, 2001, pp. 285-295.

[3] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*, ed: Springer, 2011, pp. 1-35.

[4] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence,* vol. 12, pp. 993-1001, 1990.

[5] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems (TOIS),* vol. 22, pp. 116-142, 2004.

[6] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal processing,* vol. 24, pp. 1-10, 1991.

[7] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," *arXiv preprint arXiv:1206.6426,* 2012.

[8] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing,* vol. 7, pp. 76-80, 2003.

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer,* vol. 42, 2009.

[10] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 880-887.

[11] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *Icwsm*, 2012.

[12] D. Shin, S. Cetintas, and K.-C. Lee, "Recommending Tumblr Blogs to Follow with Inductive Matrix Completion," in *RecSys Posters*, 2014.

[13] 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.

[14] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in neural information processing systems*, 2013, pp. 2643-2651.

[15] J. Wang, J. Zhou, J. Liu, P. Wonka, and J. Ye, "A safe screening rule for sparse logistic regression," in *Advances in Neural Information Processing Systems*, 2014, pp. 1053-1061.

[16] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 278-288.

[17] X. Rong, "word2vec parameter learning explained," *arXiv preprint arXiv:1411.2738,* 2014.

[18] A. Van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks,* vol. 5, pp. 465-471, 1992.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448-456.

[20] S. Wager, S. Wang, and P. S. Liang, "Dropout training as adaptive regularization," in *Advances in neural information processing systems*, 2013, pp. 351-359.

[21] M. Hanke and P. C. Hansen, "Regularization methods for large-scale problems," *Surv. Math. Ind,* vol. 3, pp. 253-315, 1993.

[22] H.-T. Lin and C.-J. Lin, "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods," *submitted to Neural Computation,* pp. 1-32, 2003.

[23] J. ZHOU, M. TANG, Y. TIAN, A. AL-DHELAAN, M. AL-RODHAAN, and S. LEE, "Social network and tag sources based augmenting collaborative recommender system," *IEICE transactions on Information and Systems,* vol. 98, pp. 902-910, 2015.

[24] S. Meyffret, E. Guillot, L. Médini, and F. Laforest, "RED: a rich epinions dataset for recommender systems," LIRIS, 2012.

[25] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Etaf: An extended trust antecedents framework for trust prediction," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, 2014, pp. 540-547.

[26] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing,* vol. 14, pp. 227-248, 1980.

[27] J. T. Townsend, "Theoretical analysis of an alphabetic confusion matrix," *Attention, Perception, & Psychophysics,* vol. 9, pp. 40-50, 1971.

[28] R. Kumar, B. Verma, and S. S. Rastogi, "Social popularity based svd++ recommender system," *International Journal of Computer Applications,* vol. 87, 2014.

[29] X. Liu and W. Wu, "Learning context-aware latent representations for context-aware collaborative filtering," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 887-890.

[30] H. Chen and A. F. Murray, "Continuous restricted Boltzmann machine with an implementable training algorithm," *IEE Proceedings-Vision, Image and Signal Processing,* vol. 150, pp. 153-158, 2003.