Linear vs. Non-Linear Embedding Methods in Recommendation Systems

1st Kerem Gürler
Research and Development Center
adesso Turkey
Istanbul, Turkey
kerem.gurler@adesso.com.tr

2nd Mustafa Coşkun
Computer Engineering Department
Abdullah Gül University
Kayseri, Turkey
mustafa.coskun@agu.edu.tr

3rd Şafak Karagenç
Research and Development Center
adesso Turkey
Istanbul, Turkey
safak.karagenc@adesso.com.tr

4th Gökhan Örün
HADİ
HADİ
Istanbul, Turkey
gokhan.orun@hadilive.com

5th Burcu Kuleli Pak
Research and Development Center
adesso Turkey
Istanbul, Turkey
burcu.kuleli@adesso.com.tr

6th Vehbi Çağrı Güngör
Department of Computer Engineering
Abdullah Gül University
Kayseri, Turkey
cagri.gungor@agu.edu.tr

2022 Innovations in Intelligent Systems and Applications Conference (ASYU) | 978-1-6654-8894-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ASYU56188.2022.9925389

***Abstract*—Predicting customer interest in items is very crucial in direct marketing as it can potentially boost sales. Data mining techniques are developed to predict which items a particular user might be interested in based on their purchase history or explicit feedback in form of ratings or comments. Recently, non-linear and linear methods have been developed for this purpose. In this study, we applied Neighborhood based Collaborative Filtering (CF), Matrix Factorization (MF), Singular Value Decomposition (SVD), Neural Graph CF (NGCF) and Light Graph Convolutional Network (LightGCN) on explicit user product rating data which is acquired from the online gaming and mobile entertainment platform called HADI. We compared the results of node embedding methods in terms of Precision@k, Recall@k and NDCG@k values. SVD and LightGCN showed the best test performance and SVD was significantly superior to LightGCN in terms of training speed. To further increase predictive performance of SVD, we have applied classification with Logistic Regression and Deep Random Forest on user and item embeddings created by the SVD.**

***Index Terms*—machine learning, deep learning, recommendation systems, node embedding, link prediction**

I. Introduction

Recommendation systems play a central role on the web to provide a personalized interaction to the web users [1]. The basic premise of a recommendation system is to estimate if a given user will like an item; by clicking, rating, and among many forms of interactions and finally they will purchase the item. Thus far, the recommendation systems remain as one of the fundamental tasks to effectively predict a user and an item interaction to personalize the user's purchase behaviours.

One such recommendation system is needed for online game platforms where a user interact with items via web advertisements. To this end, we applied various efficient and effective deep learning algorithms that can lead to a better personalized recommendation in one of the Turkish online gaming and mobile entertainment platforms, called HADI.

978-1-6654-8894-522$31.00 ©2022 IEEE

In this paper, we first give a brief background on a real-world dataset harvested from HADI game platform for creating a recommendation system. Then, we explain and compare the evaluation metrics that are developed to measure the performance of the recommendation algorithms. Subsequently, we evaluate performances of neighbourhood based CF algorithms [16], two linear embedding approaches, namely Matrix Factorization (MF) and Singular Value Decomposition (SVD), two Deep Learning based embedding models Neural Graph Collaborative Filtering (NGCF) and LightGCN [14], [15] on the HADI dataset. Models were evaluated based on precision, recall and ndcg values at top 5 recommendations. We also considered the recommendation problem as a link prediction on user-item graph and applied Logistic Regression and Deep Random Forest classifiers on user and item embedding pairs which are created by SVD method. Another embedding method which was used for link prediction named SiGraC [30] was also applied on user item graph and compared with same metrics. Novel contributions that we provide are as follows:

1) We compare recommendation performance of non-linear methods with linear methods on real world explicit feedback dataset and show that linear methods can outperform more complicated non-linear ones.
2) We further improve recommendation performance of node embeddings with making link predictions on user item graph with various models.
3) We utilize matrix approximation methods to compute SVD embeddings faster, and thus, creating an efficient and effective recommender system.

This paper is organized as follows. Related work on recommender systems with explicit data are examined in Section 2. Our dataset is explained in Section 3. Details of the methods utilized in this paper are discussed in Section 4. Results obtained for HADI dataset are presented in Section 5. Finally in Section 6, the paper is concluded.

II. Related Work

Until today, there have been various work on developing recommendation systems on explicit feedback data. Cheung et. al. [32] applied Support Vector Machine and Latent Class Model. They claim that SVM showed superior performance compared to other Content based methods and LCM model was effective in removing data sparsity problem. Gawesh et. al. [33] used both implicit and explicit feedback data and obtained similar performance results for both methods. Wang et. al. [14] proposed a Deep Learning model called NGCF that makes use of user item graph by embedding propagations on it. The method was tested on Amazon Book Review Data, Yelp dataset, which consists of user reviews of Yelp businesses and Gowalla dataset consisting of friendship network data. NGCF showed the best performance compared to their baseline methods MF, Collaborative Memory Network, Neural MF etc. He et. al. [15] proposed a new method called LightGCN which is a simplified and more linear version of GCN. It was tested on same datasets with same settings and outperformed NGCF around 16% on average. Results of their work and our tests on HADI dataset can be seen in Table 1.

TABLE I
PERFORMANCE COMPARISON OF VARIOUS METHODS ON DIFFERENT DATASETS.

Method	Dataset	Precision	Recall	NDCG
NGCF	Gowalla	-	0.157	0.132
LightGCN	Gowalla	-	0.1830	0.1554
MF	Gowalla	-	0.1291	0.1109
MF	Yelp	-	0.0433	0.0354
NGCF	Yelp	-	0.0579	0.0477
LightGCN	Yelp	-	0.0649	0.0530
MF	Amazon Book	-	0.0250	0.0196
NGCF	Amazon Book	-	0.0344	0.0263
LightGCN	Amazon Book	-	0.0411	0.0315
MF	HADI	0.3762	0.6827	0.5969
NGCF	HADI	0.3761	0.6821	0.5975
LightGCN	HADI	0.383	0.6979	0.6138
SVD	HADI	0.383	0.6965	0.6155

III. Dataset

In the online game environment, called HADI, we asked 6730 users to rate 97 items that we show in the advertisements. In this survey, depending on the users interests, they rate the items with a numeric values between 1 and 5 inclusively, where 1 represents a user is not interested in the item and 5 denotes the user is very interested in purchasing the item. To further pre-process, we remove those users who give constant number, such as 5 for each items and detect them as outliers. Then, our dataset left with 6675 users and 97 items. Subsequently, we discretized the ratings by considering above 3 ratings as interested and less or equal than 3 ratings as not interested. Finally, we divide the datasets as 80% training and 20% testing. This way, we create 259429 interactions among the user and items in the training data and 68233 interactions in the testing data. Overall, the statistics of the data is summarized in Table II.

TABLE II
DESCRIPTIVE STATISTICS OF THE NETWORKS USED IN THE COMPUTATIONAL EXPERIMENTS.

Datasets	M Users	N Items	Edges(|E|)	Train Edges	Test Edges
Hadi Dataset	6675	97	327662	259429	68233

IV. Methodology

The recommendation problem could be abstracted as link prediction problem where we want to estimate the probability of gaining an edge among given nodes in a graph [11]. Here, by threading a user and an item as two different types of node, in essence, we want to predict the link between them in a bi-partite graph. To solve the recommendation system problem, in this paper, we give four different node embedding based approaches, namely Matrix Factorization (MF), Singular Value Decomposition (SVD), Neural Graph Collaborative Filtering (NGCF), Light Graph Convolutional Network (LightGCN). With node embeddings obtained from the best performing method among listed ones and another link prediction embedding method SiGraC, we apply two classification methods for link prediction which are Logistic Regression and Deep Random Forest.

A. Evaluation Metrics

In order to evaluate state-of-the-art methods and proposed algorithm performances for recommendation system, we will use widely adopted evaluation metrics which are top-20 precision, recall, and normalized discounted cumulative gain (NDCG) whose definition can be given as follows:

i) precision@k: Correctly identified items among the k recommended items by the algorithm.

ii) recall@k: Ratio of relevant items that are recommended to user.

iii) NDCG@k: Scores hits at the top higher, therefore also taking the position of the recommendation into account.

B. Matrix Factorization (MF)

In this subsection, we first give a brief background of mathematical notations that will be used through this study for defining the recommendation systems in general. Let $\mathbf{R} \in R^{M \times N}$ be user-item matrix, where M denotes the user count and N represents the product of interest count. Let also show the non-zero entries of $\mathbf{R}$ as R and users with the letter *u* and items with the letter *i*.

In general setting, we want to learn a latent vector representation of the users an items in such a way that if the user u and the item i are close to each other in higher dimensions, $\mathbf{R}$, they must be close in the latent vector space as well. Specifically, let $\mathbf{p}_u \in R^d$ and $\mathbf{q}_i \in R^d$ be latent representations of the user *u* and the item *i*, respectively. Our objective is then to learn the latent vector representations of all items and users, $\mathbf{P} \in R^{M \times K}$ and $\mathbf{Q} \in R^{N \times K}$ such that intrinsic information in original matrix is preserved as much as possible [5].

To learn the latent representations, the early algorithms have considered the problem from the perspective of linear approaches [5] while more recent research more focus on non-linear dimensionality reductions. One such successful and linear method known as Matrix Factorization (MF), where the aim is to recover the entries of $\mathbf{R}$. Mathematically, an entry $r_{ui} \in \mathbf{R}$ can be recovered from the embedding vectors as follows [5]:

$$\hat{r}_{ui} =< \mathbf{p}_u, \mathbf{q}_i >= \mathbf{p}_u^T, \mathbf{q}_i, \tag{1}$$

Clearly, this $\hat{r}_{ui}$ can be used for matrix completion to recommend a product for a user, i.e., recommendation system. The question needed to be answered is that how we are going to learn those latent vectors, $\mathbf{p}_u$ and $\mathbf{q}_i$. To answer this, Yifan et al., [2] proposed collaborative filtering based on premise that learnt and original entries should be smoothly close each other. To this end, they give MF as following weighted regression objective function [2]:

$$J = \sum_{u=1}^{M}\sum_{i=1}^{N} w_{ui}(r_{ui} - \hat{r}_{ui})^2 + \lambda(\sum_{u=1}^{M}\|p_u\|^2 + \sum_{i=1}^{N}\|q_i\|^2) \tag{2}$$

Here, w_{ui}, represents the weight of, i.e, rating, etc, r_{ui} and $W = [w_{ui}]^{M\times N}$ is termed as weight matrix. Lastly, λ used for relaxation of the optimization problem for matrix inversion [2].

To solve the above problem, (2), one can utilize Alternating Least Square (ALS) [5], [6]. Since the equation is not convex but jointly convex, we can fix one parameter and optimize the other and visa versa [2]. To do so, in (2) we can fix $\mathbf{q}_i$ = 0 and solve J with respect to $\mathbf{p}_u$ as follows [5]:

$$J_u = \|\mathbf{W}^u(r_u - \mathbf{Q}\mathbf{p}_u)\|^2 + \lambda\|p_u\|^2, \tag{3}$$

where $\mathbf{W}^u \in R^{N\times N}$ is a diagonal matrix such that $\mathbf{W}^u_{ii} = w_{ui}$ [5]. Since the Euclidean norm is convex, in (3), derivative of J_u is taken with respect to $\mathbf{p}_u$ and it is assigned 0 to find the global minimum. [5]:

$$\begin{aligned}\frac{\partial J_u}{\partial \mathbf{p}_u} &= 2\mathbf{Q}^T\mathbf{W}^u\mathbf{Q}\mathbf{p}_u - 2\mathbf{Q}^T\mathbf{W}^u r_u + 2\lambda\mathbf{p}_u = 0 \\ &\Rightarrow \mathbf{p}_u = (\mathbf{Q}^T\mathbf{W}^u\mathbf{Q} + \lambda\mathbf{I})^{-1}\mathbf{Q}^T\mathbf{W}^u r_u\end{aligned} \tag{4}$$

where $\mathbf{I}$ is the identity matrix. With the similar logic we can attain q_i as follows [5]:

$$\mathbf{q}_i(\mathbf{P}^T\mathbf{W}^i\mathbf{P} + \lambda\mathbf{I})^{-1}\mathbf{P}^T\mathbf{W}^i r_i \tag{5}$$

C. Singular Value Decomposition (SVD)

MF method has a long standing successful history in Recommendation Systems [5]. However, its computational complexity limits its usage for large system due to cubic time complexity in its every iteration in equations (4 and 5). To avoid this cubic time complexity, we can state the same problem as an leading eigenvectors, corresponding leading eigenvalues of an adjacency matrix, computations by relying on the rich linear systems literature of computational mathematics [13].

To this end, we also applied SVD method to the recommendation system as another alternative linear approach. Mathematically, we can first give adjacency matrix of user-item graph as follows:

$$\mathbf{A} = \begin{pmatrix} 0 & \mathbf{R} \\ \mathbf{R}^T & 0 \end{pmatrix} \tag{6}$$

Now, our goal is to learn an embedding from above adjacency matrix such that learnt embedding vectors can be utilized for scoring the relationship between any given two nodes. To be more specific, let $e_u \in R^d$ be a latent vector for a user u and let $e_i \in R^d$ be a latent vector for an item i. Our aim is to learn these embedding vectors from training dataset in such a way that when we dot product them for those nodes positively paired in test dataset should result in higher values, i.e,:

$$y_{ui} = e_u^T e_i \tag{7}$$

That is, y_{ui} becomes the score to measure the closeness of nodes *u* and *i*.
When we learn the embedding, we can set an objective function that can try to recover the original adjacency matrix in equation (6). To do so, let us write the objective function as:

$$min\|\mathbf{A} - \mathbf{E}\mathbf{E}^T\|^2 \tag{8}$$

It is clear that we can use singular value decomposition (SVD) (or eigendecomposition) to solve equation(8) [13]. As an initial two linear algorithms, we implement MF and SVD and repeated them for multiple embedding dimensions and test the performance of the algorithms on HADI dataset and present the performance evaluations in Table 4 and 5.

From the table, we can see that we observe relatively high performance results for both linear methods, where SVD mostly outperforms MF. This suggests us that node embedding is highly effective in rendering recommendation system.

D. Neural Graph Collaborative Filtering (NGCF)

Wang et. al. [14] proposed NGCF model consisting of three parts in the framework which are an embedding layer that initializes user and item embeddings, embedding propagation layers which injects high order connectivity relationships and processes the embeddings, finally a prediction layer which combines the processed embeddings of different levels of propagation and results in a similarity score for a user and item.

1) Embedding Layer: For an item i, a user u and embedding vector $e_u \in R^d, e_i \in R^d$ where d shows dimension of the embedding [14].

$$E = [e_{u_1}, e_{u_2}, \ldots, e_{u_m}, e_{i_1}, e_{i_2}, \ldots, e_{i_n}] \tag{9}$$

Embedding table is initialized like shown in equation 9. In recommendation systems, like and Neural Collaborative Filtering and Matrix Factorization, this table is inputted directly to the model to calculate prediction score. NGCF model propagates embeddings on user item graph, resulting in more successful recommendation embeddings.

2) Embedding Propagation Layer: Propagation architecture is build on GNNs to be able to obtain CF signal in graph structure and optimize the user and item embeddings. Users that prefer an item can be considered as feature of that item and can be utilized to calculate similarity of two items. For an item i, we can propagate the data from its connected users in order to get the representation of e_i. Using the improved representations created with first-order connectivity, we can aggregate more layers to analyze high order connectivity.

3) Model Prediction: Embedding representations gathered from each layer is concatenated to get final embedding denoted in equation 10 as e^*. Then, we calculate user's preference score for an item as [14]:

$$NGCF(u,i) = e_u^{*T} e_i^* \tag{10}$$

E. Light Graph Convolutional Network (LightGCN)

GCNs are widely used for recommender systems. He et. al. [15] argues that two concepts in GCNs which are non-linear activation function and feature transformation increases training difficulty and reduces recommendation performance. They proposed LightGCN method which only keeps neighborhood aggregation component of GCN.

Model propagates embeddings of users and items on bipartite user item graph linearly and it takes the weighted sums of embeddings throughout the layers as the resulting embedding. GCN learns node representations by smoothing features on graph. It aggregates neighbor futures while performing iterative graph convolution and creates new representations for the target node. LightGCN does not use feature transformation and non-linear activation, and it utilizes simple weighted sum aggregator. Graph propagation in LightGCN can be shown as [15]:

$$\begin{aligned} e_u^{(k+1)} &= \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(k)}, \\ e_i^{(k+1)} &= \sum_{u \in N_i} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_u^{(k)} \end{aligned} \tag{11}$$

$\frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}}$ is symmetric normalization which is also used in standard GCN. Model training parameters in LightGCN are embeddings at the first layer e_u^0, e_i^0. Higher level embeddings are calculated with above equation. At the end of K layers, embeddings obtained by every layer are combined to create the final embedding representation [15]:

$$e_u = \sum_{k=0}^{K} \alpha_k e_u^{(k)}; e_i = \sum_{k=0}^{K} \alpha_k e_i^{(k)} \tag{12}$$

Weight of embedding at 'k'th layer is denoted as αk and can be tuned manually. Prediction is dot product of final user and item embeddings [15].

$$y_{ui} = e_u^T e_i \tag{13}$$

which constitute the scores for the recommendation rankings.

In training phase, embeddings at 0th layer E(0) are the only trainable parameters. Bayesian Personalized Ranking (BPR) loss is used as objective function [15]:

$$L_{BPR} = -\sum_{u=1}^{M} \sum_{i \in N_u} \sum_{j \notin N_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|E^{(0)}\|^2 \tag{14}$$

where λ controls the strength of regularization.

F. Link Prediction with Deep Random Forest and SiGraC

Graph structure is frequently used to show various associations and interactions among entities. Developing algorithms to analyze these graphs is one of the current research challenges. Problem of predicting new interactions utilizing the ones that exist is often called 'link prediction', a prevalent problem in machine learning [17]. Link prediction is commonly applied in biomedical network problems such as predicting drug disease associations [18], disease gene prioritization [19], drug response prediction in cancer cell lines [12], etc. Similar to biomedical network applications, we propose that problem of item recommendations can also be handled as a link prediction problem. Embedding methods are tools for link prediction task, i.e. We can use embedding vectors of the nodes as features of that node and than we can simply calculate the likelihood of gaining an edge by using these vectors dot products, hadamard product or pearson correlation as a score to identify if they are going to gain an edge or not.

Some techniques for link prediction tries to evaluate how similar the node pairs are on their topological features. These features generally concentrate on the common neighborhood of the node combinations and they can vary on evaluation of the overlap size and individual overlapping nodes. Progress in network biology showed the importance of global network topology in describing interactions between biomolecules [20]. Based on this observation, network proximity measured through random-walk based algorithms is used for the goal of link prediction [21]. Random walk methods also have limitations, such as degree bias, overstating of proximity info at the cost of structural info [22] [23], and hyperparameter choice dependency [24] [25]. Node embeddings generalize the idea of topological similarity further by creating multi dimensional latent feature representations [24] [25]. Deep learning based algorithms like Graph Convolutional Networks are also utilized to compute node embeddings [27] [28].Yue et. al.(2020) studied the performance of graph embedding methods in terms of supervised link prediction on biomedical networks and GCN based methods showed promising results [3]. An application of GCN for computing node embeddings called Graph Auto Encoder (GAE) utilizes a loss function to recreate the adjacency matrix of the interaction graph. Deep Graph Infomax (DGI) proposed by Velickovic et al. (2019) [29] limits the neural network architecture to one layer, decreasing the number of parameters that will be learned.

Coşkun and Koyutürk [30] proposes a link prediction method named Similarity-Based Graph Convolution (SiGraC)

which is applied on Drug-Drug, Drug-Disease and Protein-Protein interactions. Their method for node embedding computation utilizes DGI's single layer GCN encoder and uses node similarity matrices as convolution matrix in GCN. They compare performance of node similarity based methods with encoders which use Laplacian based methods and conclude that single layer GCN encoder outperforms Laplacian-based convolution methods.

For experimenting, we applied four different link prediction methods on our user item adjacency matrix. We generated user item embedding pairs with and SiGraC method for each positive relation in the user item graph as features and labeled them as positive. Then, we created embedding pairs for negative samples and trained a Logistic Regression and Deep Random Forest classifier for both embeddings with train test split ratio of 80/20. Finally, we compared the link prediction results of four classifiers. In order to obtain ranking metrics from classifiers, we sorted predicted probabilities of user item interactions for each user and selected the top 5 for evaluation.

V. Performance Results on HADI Dataset

We have evaluated the performance of the node embedding methods in terms of Precision@5, Recall@5 and NDCG@5 values. In addition, precision, recall and NDCG at 5 was evaluated with different embedding dimensions of 8, 16, 32, 64, 128, 256 for MF, NGCF, SVD and LightGCN.

Neighbourhood based methods were the least performing among all with metrics slightly better than randomly initialized embeddings (Table III and IX). For all three metrics, SVD (Table V) outperformed other methods. SVD showed the best performance with dimension of 32 and declined progressively at higher and lower dimensions. The performance of MF (as shown in Table III) was worse compared to other node embedding methods in terms of all metrics and dimensions. In addition to showing best evaluation performance, SVD was also significantly faster than deep learning based methods.

Speed of the algorithms is an important factor to consider because datasets can get significantly large and training times might be unfeasible for businesses. SVD showed the highest performance for both accuracy and speed on our dataset and we tested SVD approximation methods Simultaneous Iteration and Randomized Block Krylov method [31] on our data and compared both to standard SVD algorithm. Simultaneous Iteration was 14 times faster on our dataset compared to standard SVD. However, it causes a considerable drop on evaluation metrics whereas Randomized Block Krylov method was 12 times faster with insignificant drop in evaluation metrics.

Importantly, we have applied different link prediction methods on our user item adjacency matrix. Embeddings from both SiGraC and SVD showed significantly better performance with Deep Random Forest compared to Logistic Regression with SVD being slightly higher. (Table VIII) has showed the best performance with 0.753 precision@5, 0.396 recall@5 and 0.749 ndcg@5 score with balanced class distribution. Our results show that handling recommendation problem as a link prediction shows very promising results.

TABLE III
Randomly Initialized Embedding

Embedding Dimension	Precision@5	Recall@5	NDCG@5
8	0.268	0.106	0.273
16	0.264	0.105	0.268
32	0.257	0.100	0.257
64	0.266	0.106	0.269
128	0.265	0.105	0.266
256	0.266	0.105	0.268

TABLE IV
Matrix Factorization

Embedding Dimension	Precision@5	Recall@5	NDCG@5
8	0.469	0.231	0.486
16	0.461	0.227	0.482
32	0.469	0.233	0.483
64	0.483	0.242	0.491
128	0.427	0.218	0.447
256	0.474	0.234	0.478

TABLE V
Singular Value Decomposition

Embedding Dimension	Precision@5	Recall@5	NDCG@5
8	0.456	0.216	0.476
16	0.498	0.249	0.524
32	0.521	0.266	0.548
64	0.507	0.258	0.535
128	0.460	0.230	0.490
256	0.407	0.193	0.435

TABLE VI
Neural Graph Collaborative Filtering

Embedding Dimension	Precision@5	Recall@5	NDCG@5
8	0.494	0.249	0.514
16	0.472	0.234	0.508
32	0.477	0.239	0.506
64	0.481	0.244	0.510
128	0.465	0.237	0.506
256	0.463	0.236	0.503

TABLE VII
LightGCN

Embedding Dimension	Precision@5	Recall@5	NDCG@5
8	0.488	0.241	0.512
16	0.493	0.247	0.520
32	0.503	0.255	0.529
64	0.507	0.257	0.535
128	0.506	0.258	0.532
256	0.506	0.256	0.533

TABLE VIII
Link Prediction

Methods	Precision@5	Recall@5	NDCG@5
LR - SiGraC	0.403	0.176	0.415
Deep RF - SiGraC	0.720	0.361	0.749
LR - SVD	0.411	0.180	0.423
Deep RF - SVD	0.753	0.396	0.772

TABLE IX
Neighbourhood Based Collaborative Filtering

Methods	Precision@5	Recall@5	NDCG@5
Item Based	0.363	0.158	0.380
User Based	0.315	0.137	0.334

VI. Conclusion

Recently, non-linear and linear methods have been developed to predict which items a particular user might be interested in based on their feedback in the form of ratings or interactions. In this study, we have applied Neighbourhood based CF, Matrix Factorization (MF), Singular Value Decomposition (SVD), Neural Graph CF (NGCF) and Light Graph Convolutional Network (LightGCN) on explicit user product rating data, which is acquired from the online gaming and mobile entertainment platform called HADI. Comparative performance evaluations have been conducted in terms of Precision@k, Recall@k and NDCG@k values. Performance results demonstrate that SVD showed the best test performance followed by LightGCN, NGCF, MF, Item CF and User CF respectively. SVD showed slightly higher test performance than LightGCN but it was significantly superior in terms of training speed. To further increase predictive performance of SVD, we have applied classification with Logistic Regression (LR) and Deep Random Forest (DRF) on user and item embeddings created by the SVD. We also applied the same classifiers on embeddings generated by link prediction algorithm SiGraC. DRF vastly outperformed LR for both embeddings and SVD with DRF showed slightly better performance than SiGraC.

Overall, we show that non-linear models are not always superior to linear methods in the context of recommendation systems. LightGCN, which is a linearized Deep Learning model, outperformed other Neural Network models on both our study and others. SVD, which is a completely linear model, showed the best performance on our dataset with the highest test results and shorter training time compared to deep learning methods. Link prediction with DRF applied on embeddings resulted in the highest performance among all methods in our experiments, however, further research with various real world datasets should be completed to assess comparative performance of link prediction methods for recommendation systems.

Acknowledgements

This work was supported by TÜBİTAK TEYDEB Program with Project No: 3191234

References

[1] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In RecSys. 191–198

[2] Hu, Yifan, Yehuda Koren, and Chris Volinsky. "Collaborative Filtering for Implicit Feedback Datasets." ICDM. Vol. 8. 2008.

[3] Yue,X.etal.(2020).Graph embedding on biomedical networks: methods, applications and evaluations. Bioinformatics,36(4), 1241–1251.

[4] Z. Huang, D. Zeng and H. Chen, "A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce", IEEE Intelligent Systems 22 (2007), 68–78.

[5] He, X., Zhang, H., Kan, M. Y., Chua, T. S. (2016, July). Fast matrix factorization for online recommendation with implicit feedback. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 549-558). ACM.

[6] X. He, M. Gao, M.-Y. Kan, Y. Liu, and K. Sugiyama. Predicting the popularity of web 2.0 items based on user comments. In SIGIR 2014, pages 233–242

[7] Y. Koren and R. Bell. Advances in Collaborative Filtering In Recommender systems handbook, pages 145186. Springer, 2011.

[8] M. Volkovs and G. W. Yu. Effective latent models for binary feedback in recommender systems. In SIGIR 2015, pages 313322.

[9] M. Coskun, A. Grama, and M. Koyuturk, "Efficient processing of network proximity queries via chebyshev acceleration," in Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 1515–1524.

[10] M. Coskun, A. Grama, and M. Koyuturk, "Indexed fast network proximity querying," Proc. VLDB Endow., vol. 11, no. 8, pp. 840–852, Apr.2018. [Online]. Available: https://doi.org/10.14778/3204028.3204029

[11] M. Coskun and M. Koyuturk, "Link prediction in large networks by comparing the global view of nodes in the network," in Data Mining Workshop (ICDMW), 2015 IEEE International Conference on. IEEE, 2015, pp. 485–492.

[12] Z. Stanfield, M. Coskun, and M. Koyuturk, "Drug response prediction as a link prediction problem," Scientific reports, vol. 7, p. 40321, 2017.

[13] Demmel, James W. Applied numerical linear algebra. Society for Industrial and Applied Mathematics, 1997.

[14] X. Wang, X. He, M. Wang, F. Feng and T. Chua. Neural Graph Collaborative Filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Association for Computing Machinery, New York, NY, USA, 165–174. (2019)

[15] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang and M. Wang. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. SIGIR. (2020)

[16] C. C. Aggarwal. Neighborhood-Based Collaborative Filtering Recommender Systems, 29–70. (2016)

[17] Lü, L. and Zhou, T. (2011). Link prediction in complex networks: A survey. Physica A: statistical mechanics and its applications, 390(6), 1150–1170.

[18] Liang, X. et al. (2017). Lrssl: predict and interpret drug–disease associations based on data integration using sparse subspace learning. Bioinformatics, 33(8), 1187–1196.

[19] Erten, S. et al. (2011). D a d a: Degree-aware algorithms for network-based disease gene prioritization. BioData mining, 4(1), 19.

[20] Cowen, L. et al. (2017). Network propagation: a universal amplifier of genetic associations. Nature Reviews Genetics, 18(9), 551.

[21] Valdeolivas, A. et al. (2019). Random walk with restart on multiplex and heterogeneous biological networks. Bioinformatics, 35(3), 497–505.

[22] Devkota, K. et al. (2020). Glide: combining local methods and diffusion state embeddings to predict missing interactions in biological networks. Bioinformatics, 36(Supplement1), i464–i473.

[23] Ribeiro, L. F. et al. (2017). struc2vec: Learning node representations from structural identity. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 385–394.

[24] Perozzi, B. et al. (2014). Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 701–710.

[25] Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pages 855–864.

[26] Tang, J. et al. (2015). Line: Large-scale information network embedding. In Proceedings of the 24th international conference on world wide web, pages 1067–1077.

[27] Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders. arXiv preprint arXiv:1611.07308.

[28] Gilmer, J. et al. (2017). Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine LearningVolume 70, pages 1263–1272. JMLR. org.

[29] Veli ˇckovi ´c, P. et al. (2019). Deep graph infomax. 7th International Conference on Learning Representations (ICLR 2019).

[30] Mustafa Coskun, Mehmet Koyuturk, Node similarity-based graph convolution for link prediction in biological networks, Bioinformatics, Volume 37, Issue 23, 1 December 2021, Pages 4501–4508.

[31] Musco, Cameron, and Christopher Musco. "Randomized block krylov methods for stronger and faster approximate singular value decomposition." Advances in neural information processing systems 28 (2015).

[32] CHEUNG, KW, Kwok, JT, Law, MH & Tsui, KC 2000, 'Mining customer preference ratings for product recommendation using the support vector machine and the latent class model', Management Information Systems, vol. 2, pp. 601-610.

[33] Jawaheer, Gawesh & Szomszor, Martin Kostkova, Patty. (2010). Comparison of implicit and explicit feedback from an online music recommendation service. 10.1145/1869446.1869453.