

## Multi-Layer Graph Generative Model Using AutoEncoder for Recommendation Systems

Syed Falahuddin Quadri<sup>1</sup>, Xiaoyu Li<sup>1,\*</sup>, Desheng Zheng<sup>2</sup>, Muhammad Umar Aftab<sup>1</sup> and Yiming Huang<sup>3</sup>

**Abstract:** Given the glut of information on the web, it is crucially important to have a system, which will parse the information appropriately and recommend users with relevant information, this class of systems is known as *Recommendation Systems (RS)*-it is one of the most extensively used systems on the web today. Recently, *Deep Learning (DL)* models are being used to generate recommendations, as it has shown state-of-the-art (*SoTA*) results in the field of *Speech Recognition* and *Computer Vision* in the last decade. However, the *RS* is a much harder problem, as the central variable in the recommendation system's environment is the chaotic nature of the human's purchasing/consuming behaviors and their interest. These user-item interactions cannot be fully represented in the Euclidean-Space, as it will trivialize the interaction and undermine the implicit interactions patterns. So to preserve the implicit as well as explicit interactions of user and items, we propose a new graph based recommendation framework. The fundamental idea behind this framework is not only to generate the recommendations in the unsupervised fashion but to learn the dynamics of the graph and predict the short and long term interest of the users. In this paper, we propose the first step, a heuristic multi-layer high-dimensional graph which preserves the implicit and explicit interactions between users and items using *SoTA* Deep Learning models such as AutoEncoders. To generate recommendation from this generated graph a new class of neural network architecture-Graph Neural Network-can be used.

**Keywords:** Recommendation systems, autoencoder, knowledge representation, representation learning, graph-structured data.

### 1 Introduction

Information overload is one of the biggest caveats in online information systems, as an average visitor might be only interested in a tiny subset of this information. *Recommendation systems (RS)* are used to fetch these relevant subsets of information for every visitor from the huge catalog of options in real-time. The application of recommendation system is in almost every domain of web [Adomavicius and Tuzhilin

---

<sup>1</sup> School of Information and Software Engineering, University of Electronics Science and Technology of China, Chengdu, 610054, China.

<sup>2</sup> School of Computer Science, Southwest Petroleum University, 8 Xindu Avenue, Chengdu, 610500, China.

<sup>3</sup> University of Maryland Institute for Advanced Computer Studies, Bldg 224 Stadium Dr Room 3100, College Park, Maryland, 20742, USA.

\* Corresponding Author: Xiaoyu Li. Email: xiaoyu33521@163.com.

(2005)], For example, E-Commerce, Online News Platforms, Personalized Advertisements, Social Networking, Academic citations, Search Engines, and Entertainment to name a few, For this study, we will be focusing on movies recommendations. The terms *User* and *Item* is used to represent the visitor and product/service respectively.

Classically recommendation systems are classified as collaborative filtering, content-based, and hybrid recommendation systems<sup>4</sup>. Generally, a matrix  $R$ , called as *utility matrix*, is used to represent user-item preferences in a two-dimensional sparse matrix and the data is teased out of the matrix  $R$  along with additional information to calculate recommendations.

In industry, the recommendation system is no longer just a tool to save user's time and improve user experience but also important for sales of the products/services. For example, it is reported that 80% of the movies watched on Netflix [Gomez-Uribe and Hunt (2015)] and 60% of the videos watched on Youtube [Davidson, Liebold, Liu et al. (2010)] came from recommendations. Due to the direct and active interest of both industry and academia, the rate of improvement in this research area is conspicuous. The latest improvement is the use of Deep Learning (DL) methods [Zhang, Yao and Sun (2017); Covington, Adams and Sargin (2016)]. The use of DL in the field of Computer Vision and Speech Recognition has shown tremendous success in the past few decades. However, we believe a new approach-expanding the classical RS approaches-is required to fully exploit the capabilities of DL models in the realm of recommendations. This paper is a step in that trajectory.

RSs have been exceptionally improved in the last few years despite many challenges; the biggest challenge in our opinion is the Human factor, the chaotic relationship between the visitor and the information. Most of the contemporary classical RS approaches understate these chaotic variables in the system, as it is awfully difficult to fully quantize it in Euclidean space. The adoption of DL models in the RS has given us the opportunity to represent this chaotic relationship in a complex higher-dimensional space.

In this research, the latent-representations learned from the utility matrix  $R$ , using Autoencoders, are embedded with the bipartite-graph, of  $R$ , to generate a high-dimensional multi-layer knowledge-graph which can be used later to generate recommendations for the user in an online manner. The proposed generative model will be discussed in section 3.

The rest of the paper is organized as follows. Section 2 will discuss the related work, section 3 explains the proposed method. Section 4 describes the experiment. Section 5 concludes and discusses the future work.

## **2 Related work**

### ***2.1 Graph-based recommendation***

Graph-based knowledge representation has been studied for decades and it has emerged as an important model for studying complex multi-relational data, the buzzword termed by Google<sup>5</sup>, Knowledge Graph [Ehrlinger and Wöß (2016)] is mostly used in industry and academia to refer this model. This knowledge bases stores data in RDF triples (subject, predicate, object). Amazon is exploring the possibilities of building a knowledge-graph

---

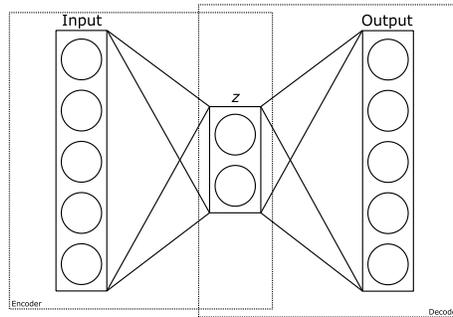
<sup>4</sup> <http://www.mmds.org/>

<sup>5</sup> <https://search.googleblog.com/2012/05/introducing-knowledge-graph-thingsnot.html>

inspired product graph<sup>6</sup> [Dong (2018)]. The graph-based approaches has been extensively researched as a link prediction problem for the recommendation systems in the literature [Chang, Harper and Terveen (2015); Cui, Zhang, Wang et al. (2016); Eksombatchai, Jindal, Liu et al. (2018); He, He, Song et al. (2018); Li and Chen (2013); Reddy, Kitsuregawa, Sreekanth et al. (2002); Zitnik, Agrawal and Leskovec (2018)]-In this models the User-Item and their relation is represented as a bipartite graph, the prediction of the missing rating is achieved by the link prediction approach. Graph-kernel based link prediction in bipartite graphs [Li and Chen (2013)], Graph-based Clustering [Reddy, Kitsuregawa, Sreekanth et al. (2002)], and Pinterest has implemented a bipartite graph based recommendation on its platform with 3+ Billions of items and 200+ Millions of user in real time.

## 2.2 Autoencoder based recommendation

**Autoencoder** is a type of artificial neural network which reconstructs the input in the output layer. It has two major components: Encoder and Decoder. In the intermediate layer (Bottleneck Layer),  $z$  in the Fig. 1 is known as latent features vector-Encoded lower dimensional representation of the input vector-and it serves as an input to the decoder. There are many variants of autoencoders and most of them can be used for recommendation task [Zhang, Yao and Sun (2017)].

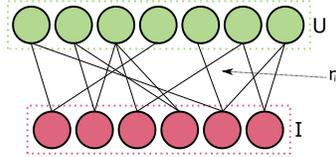


**Figure 1:** Illustration of a fully connected autoencoder with 1 hidden layer

**Collaborative Filtering and Feature Representation Learning** are successful applications of autoencoders in the recommendation systems. One of the autoencoders based recommendation model which fascinated us the most is Autorec [Sedhain, Menon, Sanner et al. (2015)], it aims to take partially observed vectors,  $r^{(i)}(r^{(u)})$ -from  $R$ -as input, project it into a low-dimensional latent space and reconstruct  $r^{(i)}(r^{(u)})$  in the output space to predict the missing ratings. The Autorec has two variants: Item-based AutoRec (I-AutoRec) and User-based AutoRec (U-AutoRec). The extension to AutoRec are *Collaborative Filtering Neural Network(CFN)* [Strub and Mary (2015); Strub, Gaudel and Mary (2016)]-uses denoising autoencoder and injects side information in the input layer-, *Collaborative Denoising Auto-Encoder (CDAE)* [Wu, DuBois, Zheng et al. (2016)]-predicts the ranking, it take the input in binary form 1 if user like the movie, 0 otherwise-, *Collaborative Deep Learning (CDL)* [Wu, DuBois, Zheng et al. (2016)], *Collaborative Deep Ranking (CDR)* [Wang, Wang and Yeung

<sup>6</sup> <http://lunadong.com/>

(2015)], *AutoSVD++* [Zhang, Tay, Yao et al. (2018)], *timeSVD++* [Koren (2010)], and clustering based outlier detection [Bo, Zhen, Qi et al. (2018)].

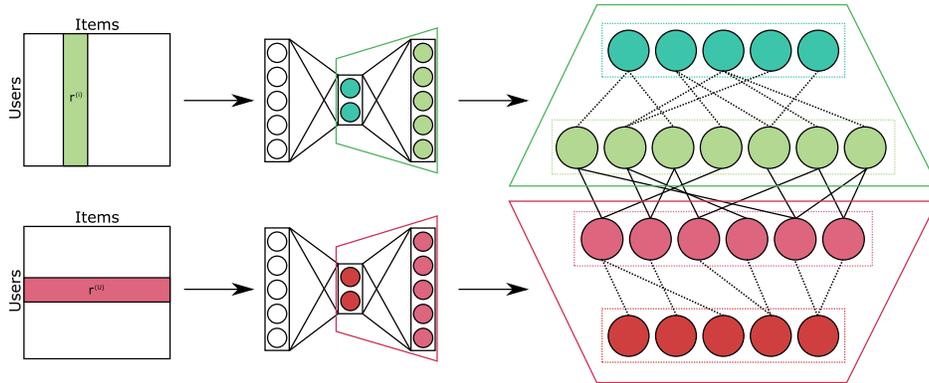


**Figure 2:** A Bipartite-Graph representation of rating matrix  $R$ , with three entities Users  $U = \{1, \dots, m\}$ , Items  $I = \{1, \dots, n\}$ , and rating  $r_{ij}$  where  $i \in U$  and  $j \in I$ . (Best-in-Colour)

### 3 Methodology

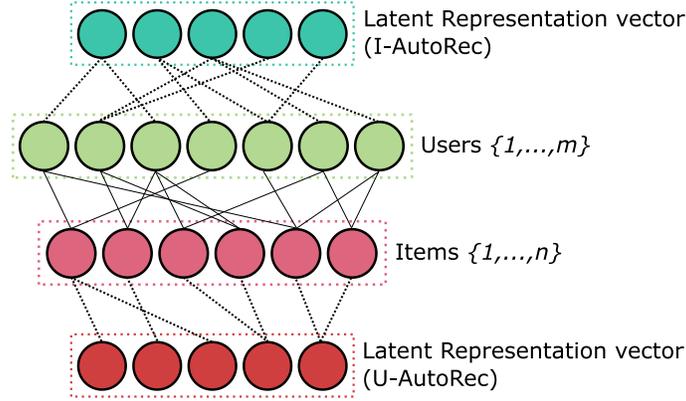
The original data-with  $m$  users,  $n$  items and a partially observed utility matrix  $R \in \mathbb{R}^{m,n}$ -is represented as a Bipartite-Graph with three entities, where  $U$  and  $I$  as the sets of nodes and elements of  $R$  as the weighted edges  $r_{ij}$  connecting  $U$  and  $I$ , as shown in Fig. 2.

Each user can be represented by a partially observed vector  $r^{(u)} = (R_{u1}, \dots, R_{un})$ , similarly for each item  $r^{(i)} = (R_{i1}, \dots, R_{im})$ , where users  $u \in U=1, \dots, m$  and items  $i \in I=1, \dots, n$  from  $R$ , as shown in Fig. 3. The sets of vectors  $r^{(u)}$  and  $r^{(i)}$  are used as the input to autoencoders, to learn the latent features. Latent features vector-bottle-neck layer-learned from the vectors  $r^{(i)}$  can be represented as the latent features representations-the top layer of Figs. 3 and 4-of the users  $u \in U$  in respect to the items  $i \in I$ , where the weights from the bottle-neck layer to the output layer are used as the weighted edges between the features representation layer to the users layer-top two layers in the Fig. 4. Similarly, the latent features vectors of the vectors  $r^{(u)}$  are represented in the graph-bottom two layers in Fig. 4. The Fig. 3 illustrates the proposed model.



**Figure 3:** Proposed Graph Generation Model using Autoencoders. (Best-in-Colour)

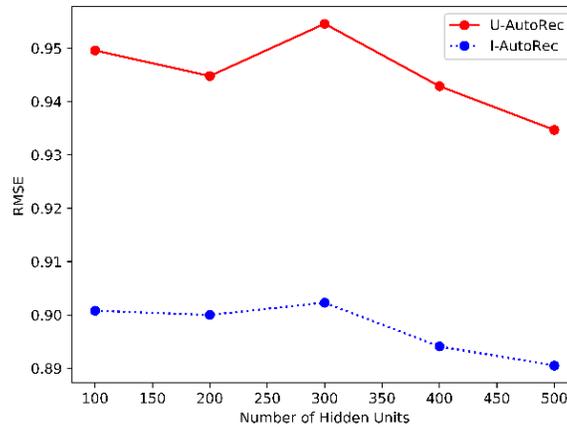
It can be noticed that the edges (dotted lines) between the learned feature layer and the entities layer are denser than the original (solid lines) bipartite-graph, as the autoencoder predicted the missing ratings. For this research, we have used only rating between user-item but additional information can be injected to make the generated graph more robust, our future efforts are in those directions.



**Figure 4:** Illustration of Proposed Multi-Layer Graph. (Best-in-Colour)

#### 4 Experiments

We have used the AutoRec [Sedhain, Menon, Sanner et al. (2015)] method to train and test the autoencoder model in both the item variant (I-AutoRec) and user variant (U-AutoRec) on the movie lens 100 K dataset [Harper and Konstan (2015)]. The data split is 70%-30% for train-test sets, the latent dimensions  $k \in \{100, 200, 300, 400, 500\}$ . Fig. 5 plots the minimum RMSE score for I-AutoRec and U-AutoRec. Blue dashed line with round markers represents the I-AutoRec variant and red solid line with round markers represents the U-AutoRec variant - in the Fig. 5.



**Figure 5:** Results of RMSE for Items-vectors and User-vectors. (Best-in-Colour)

#### 5 Conclusion and Future Work

In this paper, we proposed a novel multi-layer graph based knowledge-representation generative model-using Autoencoders. The model learns the latent-features from the utility matrix  $R$ -a sparse user-item preference matrix-then the learned latent-features are embedded with the original information to generate a high-dimensional knowledge graph, which can be later used to generate a high quality recommendation in an online manner. In

the future work, we will try to explore the possibilities of generating a finer graph which will capture all the supplementary information of the user and item using state-of-the-art embedding techniques and ways to efficiently use that graph to generate recommendations. The use of other variants of Deep Learning architectures is also a rational possibility.

**Acknowledgment:** This work is supported by The National Natural Science Foundation of China [61502082].

### References:

- Adomavicius, G.; Tuzhilin, A.** (2005): Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749.
- Bo, X.; Zhen, W.; Qi, L.; Xiaodong, L.** (2018): Smk-means: An improved mini batch k-means algorithm based on MapReduce with big data. *Computers, Materials & Continua*, vol. 56, no. 3, pp. 365-379.
- Chang, S.; Harper, F. M.; Terveen, L.** (2015): Using groups of items for preference elicitation in recommender systems. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 1258-1269.
- Covington, P.; Adams, J.; Sargin, E.** (2016): Deep neural networks for youtube recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 191-198.
- Cui, Y.; Zhang, L.; Wang, Q.; Chen, P.; Xie, C.** (2016): Heterogeneous network linkage weight based link prediction in bipartite graph for personalized recommendation. *Procedia Computer Science*, vol. 91, pp. 953-958.
- Davidson, J.; Liebald, B.; Liu, J.; Nandy, P.; Van Vleet, T. et al.** (2010): The youtube video recommendation system. *Proceedings of the Fourth ACM Conference on Recommender Systems*, pp. 293-296.
- Dong, X. L.** (2018): Challenges and innovations in building a product knowledge graph. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2869-2869.
- Ehrlinger, L.; Wöß, W.** (2016): Towards a definition of knowledge graphs. *SEMANTiCS*.
- Eksombatchai, C.; Jindal, P.; Liu, J. Z.; Liu, Y.; Sharma, R. et al.** (2018): Pixie: a system for recommending 3+ billion items to 200+ million users in real-time. arXiv:1711.07601.
- Gomez-Uribe, C. A.; Hunt, N.** (2015): The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, vol. 6, no. 4, pp. 1-13.
- Harper, F. M.; Konstan, J. A.** (2015): The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1-19.
- He, X.; He, Z.; Song, J.; Liu, Z.; Jiang, Y. G.; Chua, T. S.** (2018): Nais: neural attentive item similarity model for the recommendation. *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, pp. 2354-2366.

- Koren, Y.** (2010): Collaborative filtering with temporal dynamics. *Communications of the ACM*, vol. 53, no. 4, pp. 89-97.
- Li, X.; Chen, H.** (2013): Recommendation as link prediction in bipartite graphs. *Decision Support Systems*, vol. 54, no. 2, pp. 880-890.
- Reddy, P. K.; Kitsuregawa, M.; Sreekanth, P.; Rao, S. S.** (2002): A graph-based approach to extract a neighborhood customer community for collaborative filtering. In *Proceedings of the Second International Workshop on Databases in Networked Information Systems*, pp. 188-200.
- Sedhain, S.; Menon, A. K.; Sanner, S.; Xie, L.** (2015): Autorec: autoencoders meet collaborative filtering. *Proceedings of the 24th International Conference on World Wide Web*, pp. 111-112.
- Strub, F.; Gaudel, R.; Mary, J.** (2016): Hybrid recommender system based on autoencoders. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 11-16.
- Strub, F.; Mary, J.** (2015): Collaborative filtering with stacked denoising autoencoders and sparse inputs. *NIPS Workshop on Machine Learning for eCommerce*.
- Wang, H.; Wang, N.; Yeung, D. Y.** (2015): Collaborative deep learning for recommender systems. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1235-1244.
- Wu, Y.; DuBois, C.; Zheng, A. X.; Ester, M.** (2016): Collaborative denoising autoencoders for top-n recommender systems. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 153-162.
- Zhang, S.; Tay, Y.; Yao, L.; Sun, A.** (2018): Next item recommendation with self-attention. <https://arxiv.org/pdf/1808.06414.pdf>.
- Zhang, S.; Yao, L.; Sun, A.** (2017): Deep learning based recommender system: a survey and new perspectives. arXiv:1707.07435.
- Zitnik, M.; Agrawal, M.; Leskovec, J.** (2018): Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*.