On the use of graph embeddings instead of normal embeddings in embedding based recommender systems.

Ahmed Hussein Ragab
Computer Science Department
School of Information Technology and Computer Science , Nile University
Cairo, Egypt
ah.hussein@nu.edu.eg

Passant El-Kafrawy
ECoE, Effat University, Jeddah
School of Information Technology and Computer Science ,
CIS, Nile University, Cairo, Egypt
pelkafrawy@nu.edu.eg

***Abstract*— This paper proposes using entity2rec [1] which utilizes knowledge graph-based embeddings (node2vec) instead of traditional embedding layers in embedding based recommender systems. This opens the door to improving the performance of some of the most widely used recommender systems in the industry with just replacing the traditional embedding layer with node2vec graph embedding without the risk of completely migrating to newer SOTA systems and risking unexpected performance issues. Also, Graph embeddings will be able incorporate user and item features which can help in solving the well-known Cold start problem in recommender systems. We compare using both embedding methods on the movie-Lens 100-K dataset in an item-item collaborative filtering recommender and prove that the proposed approach improves the representation learning of the embedding layer itself which by design can improve the overall performance of the current embedding based recommenders. First, normal Recommender systems are introduced, and a brief explanation of both traditional and graph-based embeddings is presented. Then, the proposed approach is presented along with related work. Finally, results are presented along with future work.**

Keywords—Embeddings, Recommender Systems, Knowledge Graphs

I. Introduction

The advent of the world wide web gave rise to many online services that people are now using on a daily basis. From e-commerce websites like Amazon to online media streaming services like YouTube, Netflix and Hulu and even restaurant ordering apps like Door Dash, the consumer of today's world has more choices than ever. This proliferation of online services along with the technical ability to customize each user's interface differently contributed to the rise of recommender systems. Where each user is recommended a set of items (either books or other goods in e-commerce websites or movies in online streaming services, etc..) depending on his own personal preferences and previous interaction history. This allows these online platforms to deliver better customer experience , optimize their inventory and maximize their revenues all at the same time thanks to recommending non-popular items that are believed to be liked by this particular user (called long tail items) instead of recommending the same array of best seller items to all users alike.

A. Recommender Systems

Recommender systems are traditionally categorized into two main categories **Collaborative Filtering** [2], [3], [4] and **content-based** [5]. In Collaborative Filtering methods, a user-item matrix is constructed to store the interactions of each user with each item separately. Matrix factorization techniques are then used to infer from each user interactions with previous items what items he might be interested in. The main drawback of this method is its inability to handle cold-start problem which is when a new user or a new item enters the platform. On the other side, content-based methods utilize user and item features to recommend items to users even if they are new. The user features like age, gender, location, etc. are collected either directly from the user upon signing up to the platform or implied from his IP address, cookies and other online metadata. Finally, most recent SOTA recommender systems are hybrid recommenders which combine user-item interaction along with user features and item features to achieve better recommendations and in the same time overcome the cold start problem.

B. Word embeddings and their use in recommender systems

Word embeddings were first introduced for NLP tasks in [6], [7] they propelled the performance of many NLP tasks as they were able to capture the semantic similarity between words. In essence they create vectors that are semantically close to each other like the famous **Man** and **King** example illustrates that the vectors for these words are closer to each other in space and so are the words **Woman** and **Queen.**

Many recommender systems utilize this technique in creating item embeddings instead of word embedding by using the same method but replacing words with items and sentences with separate users viewing history. A detailed review and comparison of these recommenders is detailed in [8] but most notable examples are LightFM, Wide and Deep and xDeepFM. These algorithms are widely used in industry and are powering many online services today presenting millions of recommendations to users every day. Many of them are even implemented on Big data clusters and use Spark to handle the large amounts of data used. The work presented by Microsoft in [9] presents production ready implementation of many of the recommenders mentioned.

C. Knowledge Graphs

Knowledge Graph (KG) is a method used in many domains to represent large scale information including many entities and their interactions between each other [10]. Entities are represented as nodes and edges represent the relation between these nodes. If the nodes are of different types these graphs are called heterogenous graphs. Also, each node can have different attributes and the edges as well can have different weights. This approach is very useful in representing user-item interactions in recommender systems. A heterogenous graph is constructed with nodes as either users or items and edges are the interactions between them (user a bought item x or user a listened to song x). Also, if the platform has a rating system (1 to 5, like, dislike, etc..) these can be used as weights to these edges. Also, user features like age, gender and item features like artist, genre, etc. can be added. The below figure 1 from [11] illustrates a KG constructed for recommender systems.

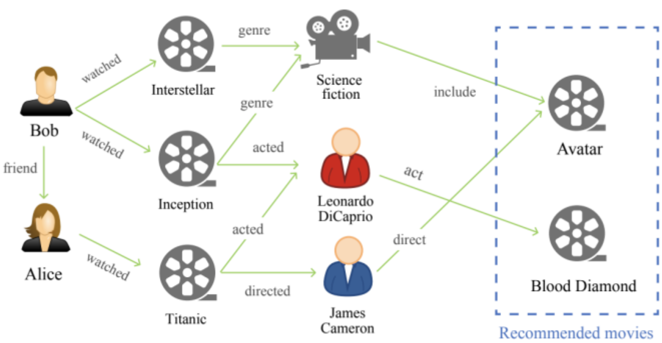

Fig.1 KG for recommender systems [9]

D. Node2vec

node2vec [12] is one of the most used task independent network representation techniques. It utilizes biased random walks in the knowledge graph to generate sequences that mimic the sentences in traditional word2vec.Then, uses these sequences to generate embeddings for these nodes that can capture deeper user-item interactions than normal embeddings.

II. Propsoed Approach

The proposed approach is to improve the performance of the widely used recommender systems that utilize traditional embedding layers with entity2rec graph embeddings to take advantage of their more powerful representation learning capabilities without much affecting the rest of the architecture in order to not affect model stability. This may be more beneficial in large scale production deployment where in some cases the marginal improvement gained from SOTA systems over current systems is not enough to offset the risk of totally revamping the recommender engine to use a totally different recommender system.

A. Replacing the normal embedding layer with graph embedding

Figures 2 and 3 illustrate the architectures of wide and deep [13] and xDeepFM [14] two of the most widely used and implemented recommender systems in production.

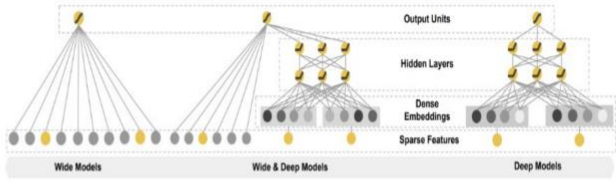

Fig.2 Wide and Deep architecture [12]

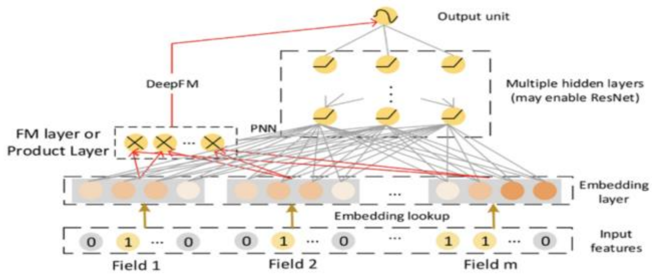

Fig.3 xDeepFM architecture [13]

B. Related work

entity2rec replaces normal embedding with Knowledge graph embedding using node2vec presents itself as an alternative and competitor to the SOTA recommenders. However, to the best of our knowledge no work has been done to validate the representation learning ability of graph embedding against normal embedding nor replace the traditional embedding layer with graph embedding layer to improve the currently deployed recommenders. Also [15] uses hierarchical Graph neural networks to present a novel architecture that specifically handles cold start problem. The work in [16] provides a Review of graph embedding based recommenders that try to out compete SOTA by proposing entirely different and novel architectures not investigating complementing currently existing ones.

III. Experiment

The aim of the experiment conducted is to asses the representation learning power of graph embedding against normal embedding. To do this , both embeddings were used on the same dataset to create a baseline item to item collaborative filtering. This method uses the item embeddings of the items that the user interacted with historically in order to predict what items the user may prefer. The dataset used is the famous MovieLens 100K [17] which presents users rating for over 9K different movies by more than 600 different users. A standard 90-10 random split was done in both cases. Also, item pairs with mutual viewership less than 10 were removed in both cases and an embedding with dimension=50 was created using both methods. Then, an item to item collaborative filtering was used to calculate the predicted rating of each user-item pair in the test using the below formula which relies on cosine similarity between movie vectors.

$$rating(U, I_i) = \frac{\sum_j rating(U, I_j) * S_{ij}}{\sum_j S_{ij}}$$

Where $rating(U, I_i)$ is the predicted rating of user U to unseen item I_i and S_{ij} is the cosine similarity between the embeddings of unseen item ***i*** and previously seen items ***j***.

IV. Results

Since the task in this dataset is to predict ratings, it is treated as a regression problem where every recommender predicts the expected rating of each user-item pair in the test set against the actual rating. The below results were extracted from the same test set in both embedding based item to item collaborative filtering recommenders.

TABLE I. NORMAL VS GRAPH EMBEDDINGS

Metric	Normal embedding	Graph embedding
MAE	0.68	**0.67**
RMSE	0.89	**0.88**

V. Conclusion

Graph embedding prove to be a worthy replacement to normal embedding layers in future recommender systems and in the same time promise to be an important improvement to currently deployed and running embedding-based recommenders without much radical changes to the remaining architecture. Upon comparing both embeddings in the same baseline recommender on the same dataset, graph embeddings were able to perform better.

VI. Future Work

several industrial recommenders like xDeepFM , wide and deep , etc.. can benefit from using graph embedding layer in replacement of normal embeddings. This represents a green field for future work of comparing the performance of both embeddings on these recommenders as well as using different datasets like lastFM for song recommendation and Amazon dataset for e-commerce.

References

[1] E. Palumbo, D. Monti, G. Rizzo, R. Troncy, and E. Baralis, "entity2rec: Property-specific knowledge graph embeddings for item recommendation," *Expert Systems with Applications*, vol. 151, p. 113235, Aug. 2020, doi: 10.1016/j.eswa.2020.113235.

[2] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan. 2003, doi: 10.1109/MIC.2003.1167344.

[3] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, Barcelona, Spain, 2010, p. 79. doi: 10.1145/1864708.1864727.

[4] X. Ning and G. Karypis, "SLIM: Sparse Linear Methods for Top-N Recommender Systems," in *2011 IEEE 11th International Conference on Data Mining*, Dec. 2011, pp. 497–506. doi: 10.1109/ICDM.2011.134.

[5] P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., *The adaptive web: methods and strategies of web personalization*. Berlin ; New York: Springer, 2007.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in Proc. of 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings, 2013.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, Red Hook, NY, USA, 2013, pp. 3111–3119.

[8] A. Ragab and P. El-Kafrawy, "Embedding Based Recommender systems, a review and comparison.," *The Egyptian Journal of Language Engineering*, vol. 9, no. 1, pp. 1–9, Mar. 2022, doi: 10.21608/ejle.2022.91884.1025.

[9] JS. Graham, J.-K. Min, and T. Wu, "Microsoft recommenders: tools to accelerate developing recommender systems," in *Proceedings of the 13th ACM Conference on Recommender Systems*, New York, NY, USA, Sep. 2019, pp. 542–543. doi: 10.1145/3298689.3346967.

[10] J. Borovina Josko, L. Ehrlinger, and W. Wöß, "Towards a Knowledge Graph to Describe and Process Data Defects," Jun. 2019.

[11] Q. Guo *et al.*, "A Survey on Knowledge Graph-Based Recommender Systems," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022, doi: 10.1109/TKDE.2020.3028705.

[12] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Aug. 2016, pp. 855–864. doi: 10.1145/2939672.2939754.

[13] H.-T. Cheng *et al.*, "Wide & Deep Learning for Recommender Systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, New York, NY, USA, Sep. 2016, pp. 7–10. doi: 10.1145/2988450.2988454.

[14] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, Jul. 2018, pp. 1754–1763. doi: 10.1145/3219819.3220023.

[15] I. Maksimov, R. Rivera-Castro, and E. Burnaev, “Addressing Cold Start in Recommender Systems with Hierarchical Graph Neural Networks,” in *2020 IEEE International Conference on Big Data (Big Data)*, Dec. 2020, pp. 5128–5137. doi: 10.1109/BigData50022.2020.9378302.

[16] Y. Deng, “Recommender Systems Based on Graph Embedding Techniques: A Review,” *IEEE Access*, vol. 10, pp. 51587–51633, 2022, doi: 10.1109/ACCESS.2022.3174197.

[17] F. M. Harper and J. A. Konstan, “The MovieLens Datasets: History and Context,” *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19:1-19:19, Dec. 2015, doi: 10.1145/2827872.