Graph Neural Networks for Recommender Systems

Heidi C. Martinsaari, Maris Häusler
Institute of Computer Science, University of Tartu

***Abstract*—150 - 200 words Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.**

***Index Terms*—Graph, Graph Embedding, Graph Convolutional Network, Recommender System.**

I. Introduction

Introduction should have background of the problem, problem description, motivation behind picking this problem, and methodology being employed, and short results summary. (approximately 1000 - 1100 words)

II. Related work

We have explored some articles about the topic of our projects. Currently we have read how our initially chosen data set of Amazon Product co-purchasing networks has been used already. First two papers below are covering this topic. We were also interested what algorithms have been used for Recommend Systems and we found lot of material about it. First three papers cover shallow algorithms. And, as our final goal is to implement Graph Convolutional Network, then half of the current papers cover different GCN topics. For the beginning, we have shortly described the papers and later we will find the connections with our project work.

A. Analysis of Product Purchase Patterns in a Co-purchase Network

Article [1] analyses the Amazon co-purchase network from the dynamic aspect. Analysis is done based on sub-category instead of item IDs for better pattern clarity. It is shown that patterns present in time t-1 are also present in time t with high probability. The authors suggest using Markov chain model on this type of data for implementing generalized recommendation system.

B. Rank the Top-N Products in Co-Purchasing Network through Discovering Overlapping Communities Using (LC-BDL) Algorithm

Paper [2] focuses on overlapping community detection algorithm based on cliques and finding top N nodes in those communities. Maximal cliques and then overlapping communities among them are identified. Top N nodes are either ranked by overlapping frequencies or by occurrence frequencies in the adjacent sub cliques that combine the discovered communities. Results obtained are evaluated using clustering coefficient and cluster density. Named techniques have also been applied on Amazon co-purchase network data.

C. entity2rec: Property-specific knowledge graph embeddings for item recommendation

entity2rec [3][4], based on node2vec [5], is tailored for Knowledge Graphs and measures user-item relatedness for top-N item recommendation. It works by creating property-specific subgraphs and finding their corresponding embeddings using node2vec. From there, similarity measures can be calculated. The advantage of entity2rec is that features can be clearly interpreted, and it works well on sparse datasets. It outperforms other commonly used collaborative filtering techniques and has slight advantage over node2vec.

D. Graph Convolutional Neural Networks for Web-Scale Recommender Systems

Paper [6] introduces GCN algorithm PinSage which combines random walks and graph convolutions to generate embeddings of nodes involving the graph structure and node feature information. PinSage is made for large scale graphs, and it is deployed at Pinterest. PinSage comes with new fundamental advancements in scalability (Big Data flavors) and introduces new training techniques and algorithmic innovations (random walks, importance pooling). PinSage results, the graph embeddings are further used in nearest neighbour lookup for recommendations or for use in a re-ranking system.

E. Knowledge Graph Convolutional Networks for Recommender Systems

KGCN [7] uses the Knowledge Graph which is usually a directed heterogeneous graph where nodes correspond to items or item attributes, while usual GCNs use mostly homogeneous graphs which nodes represent solely items. KGCNs are able to capture both the high-order structure and semantic information of the Knowledge Graph. This neural network can also learn user's current and potential interests with the help of receptive field which can be extended multiple hops away from node. KGCN uses techniques that help to train it on large scale data.

F. Structured Graph Convolutional Networks with Stochastic Masks for Recommender Systems

In the paper [8] it is pointed out that key function of GCNs is neighbourhood aggregation mechanism while real-world user-item graphs are often incomplete and noisy. It causes low performance for all graph embeddings as the aggregations are used recurrently. Authors' goal was to utilize the graph properties as the sparsity and low rank to alleviate this problem. SGCN uses NGCF and LightGCN as backbone algorithms and implements additionally the stochastic gradient mask to gain clean and sparsified graph and the nuclear norm regularization to maintain the low-rank property. Both methods appear in final loss function as regularization methods.

III. Dataset

Describe the dataset you have selected - source of the dataset, data cleaning, and some descriptive analysis if you have performed (Second lecture of the course). Link to your github account containing source code and dataset.

IV. Methodology

What is your approach (use some Figures if possible like Flowchart, to describe it).

V. Results

Here you should have some Qualitative (Visualizations), Quantitative (Tables, numbers, etc.)

VI. Conclusion

No more than 300 words.

VII. Github

- **Source code and Dataset**: Please put a link to your source code (for example, github) of your project in Report and also in the presentation. Also, provide the link to the dataset which is being used in your project. Share the exact dataset on github, which you used for your project.
- **Readme File**: Installation/Execution guide about how to run the code.

References

[1] P. Basuchowdhuri, M. K. Shekhawat and S. K. Saha, "Analysis of Product Purchase Patterns in a Co-Purchase Network," *2014 Fourth International Conference of Emerging Applications of Information Technology*, 2014, pp. 355-360, doi: 10.1109/EAIT.2014.11

[2] M. Farrag, L. Fangary, M. Nasr, and C. Salama. "Rank the Top-N Products in Co-Purchasing Network through Discovering Overlapping Communities Using (LC-BDL) Algorithm," *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, 2017, vol. 4, pp. 8153-8166

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

[4] E. Palumbo, G. Rizzo, and R. Troncy, "entity2rec: Learning User-Item Relatedness from Knowledge Graphs for Top-N Item Recommendation," *In Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17). Association for Computing Machinery*, pp. 32–36, doi: 10.1145/3109859.3109889

[5] A. Grover, J. Leskovec, "node2vec: Scalable Feature Learning for Networks," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016, doi: 10.48550/ARXIV.1607.00653

[6] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph Convolutional Neural Networks for Web-Scale Recommender Systems," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, doi: 10.1145/2F3219819.3219890

[7] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo. "Knowledge Graph Convolutional Networks for Recommender Systems," *In The World Wide Web Conference (WWW '19). Association for Computing Machinery*, 2019, pp. 3307–3313, doi: 10.1145/3308558.3313417

[8] H. Chen, L. Wang, Y. Lin, C.-C. M. Yeh, F. Wang, and H. Yang. "Structured Graph Convolutional Networks with Stochastic Masks for Recommender Systems," *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery*, 2021, pp. 614–623, doi: 10.1145/3404835.3462868