The Efficient Placement of new Products in a Co-Purchasing Graph

Markus Mueller
Business Data Science
Amsterdam, Netherlands
mueller@ese.eur.nl

Ruben Eschauzier
Business Data Science
Amsterdam, Netherlands
fill.email@email.com

Abstract

1 Introduction

for relevance, find some news articles on that

There are various reasons why a new book needs to be placed inside an existing co-purchasing graph. First, it kick-starts a recommendation system which is based on the graph structure. Also, Marketeers and others may be interested in the predicted co-purchasing links of a product currently under development. Such co-purchasing yield information about potential revenues, required Marketing efforts, direct competitors, and potential complements. Lastly, product developers may be interested in using our framework as a tool that aids their product idea generation. Our tools helps them to quickly evaluate how their product idea compares to existing customer preferences.

Besides these practical contributions, our proposed frameworks also **Markus:** list academic, technical contributions

Summary of our contributions:

- 1

2 Related Work

2.1 Placing new Product in Co-Purchasing Network

2.2 Link Prediction

Ahmed et al. (2021)

Ni et al. (2019)

Xia et al. (2021)

Zhou et al. (2020)

2.3 Clustering

The purpose of using clustering (or community detection in the context of networks) in our framework, is solely to reduce the computational burden of link prediction. By first deriving node embeddings, aligning them with the text embeddings (and other attributes), and then clustering the training data based on these node embeddings, we can first determine the cluster that reflects a new product the most before engaging in individual link prediction. **Markus:** put this part in method bit, here only general background and summary of some methods

Su et al. (2021)

Liu et al. (2020)

Hao et al. (2020)

Xu et al. (2021)

Zhang et al. (2021)

Chen et al. (2020) KNN-DBSCAM, Chen et al. (2021) BLOCK-DBSCAN. These adapt the well-known DBSCAN algorithm (REFERENCE) to make it tractable on high-dimensional data. Despite being an approximation of the original DBSCAN results, these adapted implementations still have properties which would benefit our benefit our model. First, they do not require a pre-specified number of clusters, which streamlines the training process and makes it potentially less dependent on hyperparameter choices. Second, they form clusters based on density estimation and as such can detect outliers and gather them in a separate cluster. This would allow us to potentially save computational time, if we drop the group of outliers from the latter part of our proposed framework. Unfortunately, although these adaptions achieved great first experimental results (Chen et al., 2020, 2021), they are currently still lacking an openly available implementation. Hence, we refrained from using them.

3 Methodology

Markus: put in tikz graph to explain framework and guide through sub-sections

3.1 Node Embeddings

3.2 Text Embeddings

3.3 Embedding Alignment

3.4 Clustering / Community Detection

3.5 Link Prediction

3.6 Evaluation Metrics

4 Results and Discussion

4.1 Data Description

We use Amazon metadata[1] gathered by Ni et al. (2019). These include information about a product's identifier, title, description, price, category, and co-purchasing links. Note that we consider only books and disregard other products, since books have an intuitive appeal to them that serves our proof of concept particularly well: Creating a new book to be placed in the co-purchasing network amounts to choosing a title, short description, price and a one or several (sub-) genres, i.e. its category. This reflects the actual product development framework exceptionally well and provides us with additional intuition of what our model achieves, e.g. if it places a horror-crime book in between the separate genres horror and crime in the embedding space. While this is certainly possible with other product categories, the straight-forward interpretative nature of the results may be lost in some cases. Besides that, books exhibit rather consistent descriptions and informative titles compared to other product categories, which simplifies the data preprocessing stage.

4.2 Embedding Alignment and Link Prediction

Network Settings

[1]Available `https://nijianmo.github.io/amazon/index.html#subsets`.

5 Limitations and Future Work

- extend alignment of embeddings to the inclusion of other product attributes, besides the product's ((aggregated) title and descriptions.
- using DBSCAN instead of KNN
- benchmarks for different modelling choices, e.g. only aligning text embeddings and node embeddings, instead of all attributes of a product or the link prediction model. This was unfortunately not possible with our limited computational resources.

6 Conclusions

References

Ahmed, F., Cui, Y., Fu, Y., and Chen, W. (2021). A Graph Neural Network Approach for Product Relationship Prediction. *arXiv:2105.05881*. `http://arxiv.org/abs/2105.05881`.

Chen, Y., Ruys, W., and Biros, G. (2020). KNN-DBSCAN: A DBSCAN in high dimensions. *arXiv:2009.04552*. `http://arxiv.org/abs/2009.04552`.

Chen, Y., Zhou, L., Bouguila, N., Wang, C., Chen, Y., and Du, J. (2021). BLOCK-DBSCAN: Fast clustering for large scale data. *Pattern Recognition*, 109:107624. `https://linkinghub.elsevier.com/retrieve/pii/S0031320320304271`.

Hao, Y., Cao, X., Fang, Y., Xie, X., and Wang, S. (2020). Inductive Link Prediction for Nodes Having Only Attribute Information. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 1209–1215. `http://arxiv.org/abs/2007.08053`.

Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Yang, J., and Yu, P. S. (2020). Deep Learning for Community Detection: Progress, Challenges and Opportunities. In *29th International Joint Conference on Artificial Intelligence*, pages 4981–4987. `http://arxiv.org/abs/2005.08225`.

Ni, J., Li, J., and McAuley, J. (2019). Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics. `https://www.aclweb.org/anthology/D19-1018`.

Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Jin, D., Sheng, Q. Z., and Yu, P. S. (2021). A Comprehensive Survey on Community Detection with Deep Learning. *arXiv:2105.12584*. `http://arxiv.org/abs/2105.12584`.

Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., and Liu, H. (2021). Graph Learning: A Survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127. `http://arxiv.org/abs/2105.00696`.

Xu, H., Xia, W., Gao, Q., Han, J., and Gao, X. (2021). Graph embedding clustering: Graph attention auto-encoder with cluster-specificity distribution. *Neural Networks*, 142:221–230. `https://linkinghub.elsevier.com/retrieve/pii/S0893608021002008`.

Zhang, X., Liu, H., Wu, X.-M., Zhang, X., and Liu, X. (2021). Spectral embedding network for attributed graph clustering. *Neural Networks*, 142:388–396. `https://linkinghub.elsevier.com/retrieve/pii/S0893608021002227`.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81. `https://linkinghub.elsevier.com/retrieve/pii/S2666651021000012`.