Boosting Recommender Systems with Advanced Embedding Models

Gjorgjina Cenikj
Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, Republic of North Macedonia
gjorgjina.cenikj@students.finki.ukim.mk

Sonja Gievska
Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, Republic of North Macedonia
sonja.gievska@finki.ukim.mk

ABSTRACT

Recommender systems are paramount in providing personalized content and intelligent content filtering on any social media platform, web portal, and online application. In line with the current trends in the field directed towards mapping problem and data encoding representations from other fields, this research investigates the feasibility of augmenting a graph-based recommender system for Amazon products with two state-of-the-art representation models. In particular, the potential benefits of using the language representation model BERT and GraphSage based representations of nodes and edges for improving the quality of the recommendations were investigated. Link prediction and link attribute inference were used to identify the products that the users will buy and predict the rating they will give to a product, respectively. The initial results of our exploratory study are encouraging and point to potential directions for future research.

CCS CONCEPTS

• **Theory of computation** → Design and analysis of algorithms; Graph algorithms analysis; • **Computing methodologies** → Artificial intelligence; Natural language processing; Information extraction; Machine learning; Machine learning approaches; Neural networks; • **Applied computing** → Electronic commerce; Online shopping.

KEYWORDS

recommender systems, link prediction, graph embeddings, word embeddings

ACM Reference Format:
Gjorgjina Cenikj and Sonja Gievska. 2019. Boosting Recommender Systems with Advanced Embedding Models. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3366424.3383300

1 INTRODUCTION

A critical requirement for advancing the relevance and usefulness of information in intelligent interactive systems pertains to omnipresent recommender systems. The limitations of collaborative filtering (e.g., scalability, data sparsity, and popularity bias) and content-based filtering (e.g., excessive specialization, loss of serendipity, and pigeonholing) have led to intensive scientific investigation for meliorating these challenges. Current research efforts in the field are directed towards improving the efficacy and scalability of the existing algorithms [2, 5, 9], extending the set of contextual factors on which the recommender systems rely upon [7], mapping modeling and algorithmic approaches from other domains [5], exploring advantageous problem representations and metadata encoding strategies [3, 4]. Recent research highlights the benefits of inclusion of user generated data, such as product reviews, comments or tags [7] by providing insights into user's preferences, past interactions and behavior.

The current predominance of deep learning for processing abundant textual information complemented by distributed vector representation of words, sentences and documents have recurrently proven their superiority on a number of tasks, including recommender systems [7]. Conversely, mapping network structure in a low dimensional vector space offered by ever-sophisticated graph embeddings [3, 4] has found their place in graph-based recommender systems.

Graph Convolutional Networks (GCN) [10], which aggregate feature information from local graph neighborhoods and utilize both the network structure and node attributes are one of the most prominent in the field. The motivation behind this research was to utilize a combination of state-of-the-art approaches to build a graph-based recommender system for Amazon products. User product associations implied by user's ratings and reviews given to products were used for construction of a bipartite user-product interaction graph. Node embeddings generated by GraphSage are used to represent the graph data, while BERT embeddings are used to represent textual information. Link prediction is used for recommending products i.e., making predictions whether a given user will buy a certain product, and link attribute inference is used to predict the rating that the user will give to a product. The Amazon dataset, which contains explicit ratings on a scale from 1 to 5, raw text of user reviews and product metadata was used to evaluate the proposed models.

After a short review of the most related research in Section 2, a detailed description of the proposed models are explained in Section 3. The discussion of the evaluation results are presented in Section 4, while Section 5 concludes the study.

2 RELATED WORK

This research draws upon a number of approaches that have been previously fruitfully applied on tasks with related objectives. A study investigating the effect of different node embeddings on

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3383300

the task of link prediction in biomedical graphs of realistic sizes has been presented in [3].Concatenation of node embeddings as a means of representing current and potential edges in the graph has been proposed. The system predicts whether each edge is likely to exist or not by feeding the edge embeddings into a simple neural network that outputs the probability of an edge forming between the nodes. The performance of 4 graph embedding methods (DeepWalk, Node2Vec, LINE, and SDNE) are compared on 3 biomedical tasks, predicting drug-target interactions, protein-protein interactions and links among co-occurring biomedical entities in medical publications. While the objective of this research differs from ours, interactions in biomedical graphs as opposed to product recommendation, and we opted for different types of embeddings, we found this work inspirational and useful when making decisions in our work.

Node embeddings tailored to exploit edge heterogeneity in multi-graphs have been proposed for recommending friends on the Hike Network. [4] DeepWalk and Node2Vec algorithms were considered as options for generating node embeddings. Three extensions of these algorithms are proposed for heterogeneous multi-graphs. Additionally, averaging, concatenation, and Hadamard product were explored as strategies for combining the node embeddings into edge embeddings. In our research, GraphSage algorithm was adopted for generation of node embeddings in a homogeneous and bipartite graph, while edge embeddings were created by taking their inner product.

A number of studies have pointed out that enhancing recommender systems by fusing diverse in nature features is a promising direction for improving and diversifying recommendations [1]. A joint use of network-related characteristics, node embeddings and natural language processing of user profile and user generated text has been shown to overcome the shortcomings of traditional models for subreddit recommendation [1]. Word2Vec and Node2Vec have been used for generating text and node embeddings, respectively. In contrast, our approach employs more recent advances, namely, the GraphSage algorithm for generating node embeddings, and the state-of-the-art BERT algorithm for text embeddings.

PinSage [9] is a highly scalable GCN algorithm based on random walks and graph convolution that is capable of generating node embeddings of graphs containing billions of nodes. A number of data-efficient strategies have been employed, from low-latency random walks to sample graph neighborhoods. The paper also introduces a highly efficient MapReduce pipeline that provides the high scalability of the algorithm. PinSage has been applied on the Pinterest dataset, in order to recommend pins to users. The inspiration drawn from this work mainly relates to modeling the problem at hand as a bipartite graph of disjoint sets of nodes representing products and users. The language representation model BERT was our choice for representing the textual information, as opposed to Word2Vec. The incorporation of visual features was not applicable on our dataset.

3 METHODOLOGY

A founding point for the graph-based recommender advanced in this research was the creation of a user-product interaction graph based on the assumption that user preferences and interests are implied in user's ratings and reviews of products. Vector representations i.e., embeddings in the graph based recommender system presented in this research are used at two levels: graph i.e. node and edge embeddings are used to represent information in the product-user interaction graph generated on the basis of the Amazon dataset, while the language representation model BERT is used to represent textual information related to products, reviews and users.

BERT is a state-of-the-art algorithm used in natural language processing. Its main technical innovation is the Transformer, an attention model which reads an entire sequence of words at once, meaning that it is bidirectional and can learn the context of a word from its surroundings [6]. GraphSage is a framework for representation learning in large graphs, which leverages node feature information to generate node embeddings for previously unseen data. As opposed to training individual embeddings for each node, it inductively learns a function that generates embeddings by sampling and aggregating features from a local neighborhood [11].

Two recommendation models are proposed. The model based on link prediction aims to anticipate the products that each user will buy, while the model based on link attribute prediction aims to predict the rating that the user will give. Link prediction aims to infer missing edges or predicting future ones based on currently observed partial networks. Two types of link prediction techniques are distinguished: proximity-based and learning-based methods. Proximity-based methods are used in practice due to their simplicity and interpretability, but since they rely heavily on strong assumptions on when two nodes are likely to link, they are not effective on networks where these assumptions fail. In this regard, a more reasonable way should be learning a suitable heuristic from a given network instead of using predefined ones [8]. The learning-based approaches automatically learn the weights, which are going to be assigned to the input features. The link prediction problem is modeled as a binary classification, where, given a pair of nodes, the goal is to determine whether they belong to a positive or a negative class ("positive" meaning that there is an edge between them, and "negative" meaning otherwise).

3.1 Dataset

The complete Amazon dataset[1] reviews collected in the period from May 1996 to July 2014. A subset containing 41,252 product reviews, from 3244 unique users on 2412 unique office products has been selected for the purpose of evaluating the proposed recommender models in the study. Each data sample contains a user id, a product id, the date the review was made, the raw text of a review, the scores assigned by a user, product ratings in the range 1-5, and a helpfulness score. Supplemental information for each product, such as: description, price, categories, title, sales rank, related products and brand were also provided in a separate dataset. This information is referred to as product metadata.

3.2 Method

3.2.1 Graph construction. An undirected, bipartite, weighted graph was constructed based on the product and review data from the Amazon dataset with nodes representing users and products. An edge between a user-product pair was generated if the user has

[1] http://jmcauley.ucsd.edu/data/amazon/contains142.8million

reviewed the product and a weight corresponding to the user's rating of a product was associated with each edge.

3.2.2 *Generation of node features.*

Initial preprocessing was performed on the raw text of the reviews and the description of each product by converting them to lowercase and removing punctuation. Three types of vector representations that differ in the type of product-related features they encode were examined and evaluated:

- Category&Brand: A one-hot representation of the products' category and brand information.
- ProductDescription: Maps the products' description into low-dimensional vector space using a pre-trained BERT model, which derives semantically meaningful word embeddings of size 768 using the second to last layers, which are then averaged to form document embeddings.
- Review&ProductDescription: Two pre-trained BERT representations were created, one corresponding to product description and the other to the raw text of the reviews. The embeddings were later averaged to generate the resulting vector for a particular product.

In the absence of user-related information in the dataset, an approximate representation of each user was generated by averaging the vectors of all products they have reviewed. By doing this, the dimensionality of both types of nodes, were made identical.

3.2.3 *Generation of node embeddings.*

By using the GraphSage model [5] low-dimensional vector representation of the nodes in the initially created graph of user-product interactions were generated. The model takes user product pairs of nodes as inputs and outputs their corresponding node representations. The assumption that nodes that reside in the same neighborhood would have similar embeddings introduces the need for aggregation functions, which combine each neighbor's embedding with weights to create a neighborhood embedding, i.e. aggregated information from the node's neighborhood. When the aggregator weights are learned, the embedding of an unseen node can be generated from its features and neighborhood, hence, removing the need for re-training when new nodes are introduced to the graph.

3.2.4 *A recommender model based on link prediction.*

The GraphSage-generated node vector representations were fed into a link classifier, which first constructs the edge embeddings by calculating the inner product of the node embeddings. A fully connected layer is used as a binary classifier to obtain the probability that each edge exists in the graph. The models are trained using the Adam optimizer, a binary cross-entropy loss, a learning rate of 0.001 and an early stopping strategy.

3.2.5 *A recommender model based on link attribute inference.*

Since the graph is modeled in such a way that the given ratings are used as the weights of the edges connecting a user and a product, the problem of predicting the rating that a user will give to a product can be treated as a link attribute inference. Such a model can easily be obtained by replacing the link classifier layer in the link prediction model with a link regression layer, which outputs a weight for each edge, i.e. the predicted rating a user would give to a product.

4 DISCUSSION OF RESULTS

The test dataset is created by splitting the dataset chronologically, namely 2014 was used as a splitting point, so that the prediction of buying products after 2014 was based upon users' purchase history before 2014. The test dataset contains 23% of the original reviews, and the train dataset contains 77%. Since each edge in the graph indicates that a user has reviewed a product, this procedure refers to the generation of the positive samples in each of the datasets. The negative samples are obtained by randomly sampling an equal number of non-existent edges in the graph as the number of positive samples, so that we end up with balanced train, test and validation datasets.

4.1 Evaluation of the recommender model based on link prediction

ROC AUC (Receiver Operating Characteristic - Area Under the Curve) and the AP (average precision) metrics were used to evaluate the performance of the model based on link prediction. The ROC AUC measures the entire two-dimensional area underneath the entire ROC curve, which plots the True Positive Rate and the False Positive Rate, while the AP summarizes a precision-recall curve as the weighted mean of precision achieved at each threshold, with the increase in recall from the previous threshold used as the weight. Our own implementation of the random walk algorithm for link prediction was used as a baseline model against which the link prediction based approach for recommender was compared; three versions, one for each type of node representation.

Table 1 presents the evaluation results and shows that the model proposed in this paper outperforms the results of the baseline model, regardless of the type of representation used. The three representations achieve comparable performance, with the Review&ProductDescription yielding the best results. The Category&Brand representation has yielded a slightly better average precision by order of 0.01 when compared to the best performing Review&ProductDescription, although it should be noted that the Review&ProductDescription representation is more computationally demanding, and required 5 more hours to generate the review embeddings.

4.2 Evaluation of the recommender model based on link attribute inference

RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) were used to evaluate the performance of the models on the rating prediction task shown in Table 2. RMSE is a measure calculated as the square root of the average of the squared differences between the predicted rating and the actual rating, while MAE is calculated as the average of the absolute differences between the predicted rating and the actual rating.

Our own application of the popular Singular Value Decomposition (SVD) matrix factorization based algorithm from the Surprise library, and a custom implemented content based filtering algorithm (CBF), which uses cosine similarity of product vectors were used as baseline models that were trained and tested on the same dataset.

The proposed recommender model based on link attribute inference outperforms the CBF algorithm for all three types of node

Table 1: Evaluation results of the recommender model based on link prediction for the baseline model based on random walks and proposed three node representations

Representation	AUC-ROC	AP
Random walk	0.78	0.78
Category&Brand	0.86	0.82
ProductDescription	0.85	0.81
Review&ProductDescription	0.87	0.83

Table 2: Evaluation results of the recommender model based on link attribute inference for two baseline models, Singular Value Decomposition (SVD) and Content-based Filtering (SBF), and the proposed node representations

Representation	RMSE	MAE
SVD	0.87	0.68
CBF	0.96	0.74
Category&Brand	0.90	0.70
ProductDescription	0.91	0.70
Review&ProductDescription	0.94	0.70

representations, although fails to reach the performance of the SVD algorithm.

The model with the best performing Category&Brand representation differs in 0.03 and 0.02 orders of magnitude in the RMSE and MAE metrics, respectively. The slight performance difference could be attributed to the relatively small dataset used in our experiments, having in mind that deep learning models are better suited for larger quantities of data. Increasing the dataset might also allow us to use representations of higher dimensionality, i.e. fusion of the features of two or more of the proposed node representations that might be expected to improve the performance.

Surprisingly, encoding the text of user reviews incorporated in the third representation for the product nodes, Review&ProductDescription, yields the weakest results out of the three suggested representations. This might indicate that we rely too much on our assumption that user reviews reflect user's preferences and interests.

A potential idea worth exploring is using a weighted average of BERT review vectors for vector representation of both types of nodes, user and products, where the weights would be the normalized helpfulness scores of each review. Our current efforts are directed toward employment of dimensionality reduction methods to all types of vector representations that were used.

Future work could be directed into utilizing the attributes "also bought' or 'also viewed' that are part of the product metadata of the Amazon dataset, but were not taken into account for the node embeddings. Two alternatives might be fruitfully explored. One option would be augmenting the graph with heterogeneous type of edges, although by adding a product-product edge for related products based on those attributes will result in losing the graph's bipartite structure. Another possibility would be to generate two graphs, one representing the user-product and the other product-product associations.

Another idea for possible improvements would be to incorporate additional sentiment analysis models and use a heterogeneous bipartite graph where for each review, there would be different types of edges with weights corresponding to the rating and the sentiment analysis score.

5 CONCLUSIONS

In this paper, we examine the impact of the latest advances in language and graph representation models to a recommender system for Amazon products. Link prediction and link attribute inference were considered as techniques for predicting the customers' buying and rating patterns. While link attribute inference did not yield performance advantage on the dataset used in our study, the empirical evidence has noteworthy implications for future research. Employing link prediction for recommendation of product has obtained more appealing results and certainly seems like a promising approach worth dedicating more research efforts to.

ACKNOWLEDGMENTS

This work was partially financed by the Faculty of Computer Science and Engineering at the "Ss. Cyril and Methodius" University.

REFERENCES

[1] Andrej Janchevski and Sonja Gievska. 2019. A Study of Different Models for Subreddit Recommendation Based on User-Community Interaction. In International Conference on ICT Innovations (2019), 96-108 DOI: 10.1007/978-3-030-33110-8_9
[2] Christos Sardianos, Grigorios Ballas Papadatos, and Iraklis Varlamis. 2019. Optimizing Parallel Collaborative Filtering Approaches for Improving Recommendation Systems Performance. Information. 10. 155. (2019) DOI: 10.3390/info10050155
[3] Gamal Crichton, Yufan Guo, Sampo Pyysalo and Anna Korhonen. 2018. Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches. BMC Bioinformatics 19, 176 (2018) DOI: 10.1186/s12859-018-2163-9
[4] Janu Verma, Srishti Gupta, and Debdoot Mukherjee. 2019. Heterogeneous Edge Embeddings for Friend Recommendation. Advances in Information Retrieval. ECIR 2019. Lecture Notes in Computer Science, vol 11438. Springer, Cham (2019)
[5] Marwa Hussien Mohamed, Mohamed Helmy Khafagy and Mohamed Hasan Ibrahim. 2019. Recommender Systems Challenges and Solutions Survey. 2019 International Conference on Innovative Trends in Computer Engineering (ITCE'2019), Aswan, Egypt (2019) DOI: 10.1109/ITCE.2019.8646645

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova.2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019, pages 4171–4186

[7] Pasquale Lops, Dietmar Jannach, Cataldo Musto, Toine Bogers, and Marijn Koolen. 2019. Trends in content-based recommendation. User Model User-Adap Inter 29 (2019), 239–249 DOI: https://doi.org/10.1007/s11257-019-09231-w

[8] Poorna Kumar, Amelia Lemionet and Viswajith Venugopal. 2016. Link Prediction in the Pinterest Network. (2016)

[9] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018) 974-983 DOI: 10.1145/3219819.3219890

[10] Si Zhang, Hanghang Tong, Jiejun Xu and Ross Maciejewski. 2019. Graph convolutional networks: a comprehensive review. Computational Social Networks 6, 11 (2019) DOI: https://doi.org/10.1186/s40649-019-0069-y

[11] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs.NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems (2017), 1025-10 DOI: https://arxiv.org/abs/1706.02216