Nathan Tsai & Abdullatif Jarkas

Professors Mishne & Fraenkel

DSC 180B

7 March 2021

Project Report

Graph-based Recommender System

Abstract

Recommender systems are important, revenue-generating technologies in many of the services today, providing recommendations for social, product, and other networks. However, the majority of existing recommender system methods use metrics of similarity to recommend other nodes through content-based and collaborative filtering approaches, which do not take into account the graph structure of the relationships between the nodes. A graph-based recommender system then is able to utilize graph relationships to improve node embeddings for recommendation in a way that conventional recommender systems cannot. Inspired by PinSage[1], we explore an unsupervised graph-based recommendation method that can take advantage of the relationships between nodes, in addition to the text and image features, and generate more accurate and robust embeddings for Amazon product recommendation.

Introduction

Recommender systems are responsible for large revenues and consumer satisfaction in many of the services used today. Widely-used services, such as Netflix, Facebook, Amazon, and LinkedIn, use recommender systems to suggest movies, posts, users, and products to their consumers. Traditional recommender system methods use metrics of similarity to recommend other products through content-based and collaborative filtering approaches. However, product data can be expressed in a non-Euclidean graph format with relationships, such as products bought together or products viewed together. These recommender system methods do not take into account the graph relationships between the product nodes to improve recommendations. A graph-based approach to recommendation is able to fully utilize the relationships between product nodes, in addition to any product text and image features, to generate more accurate and robust embeddings, compared to embeddings from traditional recommender systems.

Related Work

Our work builds upon the existing advancement in applying graph neural networks to recommender systems. Graph convolutional networks (GCNs)[2] have allowed deep learning to harness the power of non-Euclidean data, providing relationship and structure data to deep learning techniques. GraphSAGE[3] introduced an inductive approach to generating embeddings that sampled neighboring nodes and aggregated their features to produce embeddings. PinSage[1] improved upon the GraphSAGE algorithm by introducing a graph-based recommender system with a new sampling and aggregation process and providing an efficient technique for large, web-scale training for production models. We adapt the PinSage[1] algorithm to work in an unsupervised learning context to generate more robust and accurate product embeddings that take into account the underlying product graph structure.

Methods

Dataset

The product dataset used is the Amazon Product Reviews dataset from Professor Julian McAuley. Each product has image and CNN vector features that can be used as node features, and edges are provided or can be generated using spectral clustering techniques. Edges between product nodes are expressed by products that were also viewed or bought by users. The image features combined with the text features can also be passed through a CNN to generate an image and text embedding that can also be used as a feature in graph learning.

Cleaning & Data Ingestion Process

In our dataset we are given Product IDs (ASIN) and related products (product IDs that are also viewed and bought). Our model attempts to predict related products within the graph, such as also viewed and also bought products. However our data did not come all clean and complete as we needed. Our data had a lot of missing entries such as titles, names, publishers, related, and image sources. So, we dropped all the missing rows and kept the ones that are full and complete. We then had to go into the related products (also viewed and bought products) and remove all product IDs that were not in our dataset. So now we are left with 24,012 products (nodes), all with related products available (edges), full with text and image features forming a fully connected graph.

As our model requires image and text features to work, we have to prep those as well. Image features have been already provided by the McAuley so we don't need to do much from there however, worst case we can manually download them through the image source link provided in our dataset and convert them to tensor objects. As for text data we will be using the TorchText library to build text features such as title, and description from a vocabulary set.

Model

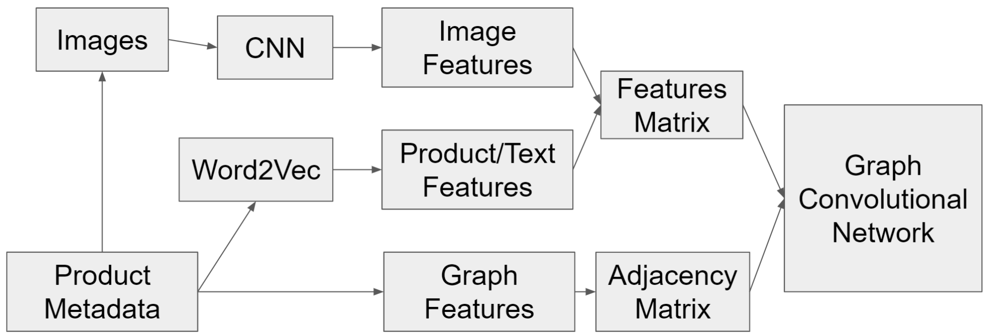

Figure 1. Overview of Model and Data Pipeline.

Our model is based on PinSage[1] and learns to generate node embeddings using visual features, textual features, other product features, and graph features. Visual features are image embeddings generated by passing product images through a deep convolutional neural network, as described in [4]. Textual features are word embeddings generated by training a Word2Vec-based model[5].

Nearest Neighbor Recommendation

By combining all three embedding approaches, we can get a more complete and robust embedding of the product for recommendation. The closest embeddings to the generated embedding will be the most related products and be recommended by model. Using locality sensitive hashing methods[6], the nearest product embeddings will be calculated and recommended.

Results

TBD.

Discussion

TBD.

References

[1] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems.

[2] T. N. Kipf and M. Welling. 2017. Semi-supervised Classification with Graph Convolutional Networks.

[3] W. L. Hamilton, R. Ying, and J. Leskovec. 2017. Inductive Representation Learning on Large Graphs.

[4] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel. 2015. Image-based Recommendations on Styles and Substitutes.

[5] T. Mikolov, I Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality.

[6] A. Andoni and P. Indyk. 2006. Near-optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions.

Appendix

Project Proposal

Problem & Data

This project is to examine recommender systems using a graph-based learning approach. Recommender systems are responsible for large revenues and consumer satisfaction in many of the services used today. Widely-used services, such as Netflix, Facebook, Amazon, and LinkedIn. use recommender systems to suggest movies, posts, users, and products to their consumers. This project focuses on the application of graph learning to augment product recommendation. Current recommender system methods use metrics of similarity to recommend other products through content-based and collaborative filtering methods. However, product data can be expressed in a non-Euclidean graph format with relationships, such as products bought together or products in similar categories. These recommender system methods do not take into account the relationships between the product nodes, not utilizing the graph structure of data to improve recommendations. Graph-based learning can take advantage of the relationships between product nodes, in addition to the product text and image features, and generate more accurate and robust embeddings. A graph-based recommender system then is able to utilize improved product embeddings to recommend products, utilizing graph data in a way that conventional recommender systems cannot.

The dataset used in the project will be the Amazon Product Reviews dataset from Professor Julian McAuley (https://nijianmo.github.io/amazon/index.html). Each product has image and CNN features that can be used as node features, and edges are provided or can be generated using spectral clustering techniques. Edges between product nodes are expressed by products that were also viewed or bought by users. The image data can also be passed through a CNN to generate an image embedding that can also be used as a feature in graph learning.

Once we've developed a graph recommender algorithm for Amazon products we may potentially get into graph classifications concentrating on the Amazon user and their features. For example, our dataset contains product review data which contains data on both the product and the user. We could use this data with minor data ingestion and processing techniques to create nodes on users with node features such as products reviewed, products viewed, and etc. We can use this data to find and categorize different users by their niche interests through these different node features. In this case we'd be creating an algorithm similar to graph classifications on social media networks, but on Amazon customers and their products. Which would be a unique approach to user graph classifications.

Methods

The images are provided within the dataset as URLs, the descriptions and other text features are also provided in the dataset, and the graph relationships between nodes will be constructed from the provided IDs of products also bought or viewed by similar users. We plan to utilize a convolutional neural network to generate an image feature embedding from the product image, a neural network model like word2vec to generate a text feature embedding, and a graph neural network like graphSAGE or node2vec to generate graph embeddings. By combining all three embedding approaches, we can get a more complete and robust embedding of the product for recommendation. The closest embeddings to the generated embedding will be the most related products and be recommended by model.

Project Output

The output of the project will be a report that describes how graph-based learning approaches can be used with recommender systems and include comparisons with the various recommender system approaches on benchmark datasets. The report will include examples of the recommendations across models given the same product input. An optional website output can focus on returning the images of the top recommended products within the dataset based on product information that the user submits.