Category Classification for Amazon Items Using Hidden State Node Embeddings of Large Graphs

Fengjiao Lyu
fengjiao@stanford.edu

Joseph Lee
wejlee@stanford.edu

Yaqing Li
yaqing2@stanford.edu

I. Abstract

In recent years, huge advances have been made in both Natural Language Processing (NLP) techniques as well as representation learning on graphs. However, to our knowledge, these two fields remain largely distinct. We thus aim to synergize these two fields by employing an original and novel technique of using the hidden states from NLP techniques (CNN, RNN (with GRU/LSTM)) as node embeddings for our inductive graph techniques (GraphSAGE). We applied this novel technique to predicting which category an Amazon item is classified under. This can help suggest likely categories and genres for new items listed. On this difficult task, we achieved an accuracy of 0.410, improving on our baseline by 54% - 76%. In addition, we experimented with non-uniform sampling which has improved the existing GraphSAGE algorithm on its reported baseline.

II. Introduction

Graphs are very useful in storing structured knowledge such as network data, social networks and knowledge graphs. These graphs often provide useful information for modern machine learning applications such as link prediction, node classification, and clustering [1]. It is thus of no surprise that the field of applying Machine Learning techniques to handle graph data has been steadily growing with new algorithms such as Graph Convolutional Networks, GraphSAGE and Graph Attention Networks in just the past year.

However, while the focus on these graph algorithms have been on how to cleverly use the graph structure to optimize learning, the features they use as node embedding inputs remain rather unoriginal. The current methodologies do not tap into the vast improvements in NLP techniques to augment the input features of these graph algorithms. We thus wish to combine cutting-edge NLP techniques with advances in graph techniques to explore the accuracy gains from synergizing these two fields which are currently distinct from each other. To our knowledge, this is a direction that has not been explored by researchers yet.

We wanted to test our hypothesis on an application area which was significant and could improve the lives of users. We identified category classification for items listed as Amazon.com as our application area, since this system could automatically suggest to users likely categories for new items (such as new books or CDs). This saves users time as they no longer have to trawl through a list of more than a hundred different possible categories. However, we do stress that while we have applied this technique to the Amazon dataset, this new framework is also relevant in many other contexts involving social and economic networks.

Lastly, we noticed that GraphSAGE employed uniform sampling during their node sampling process. We were interested to see if cleverly sampling neighboring nodes based on the graph information (such as the degree of the nodes) would provide algorithmic improvements to the original GraphSAGE algorithm. We thus tested a few different sampling methods on the Reddit dataset (the dataset used in the original paper) as well as on our new application area of the Amazon dataset.

III. Related work on Machine Learning on Graphs

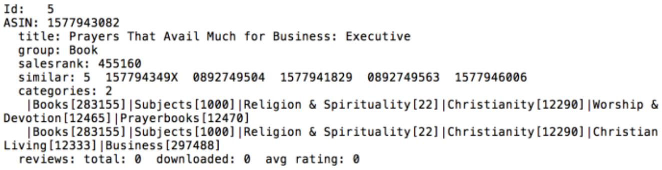
Recent advances in representation learning on graphs using convolutional learning methods have achieved state-of-the-art results on tasks like link prediction and node classification [2-4].

Most of the studies on Machine Learning applied to graph data such as Graph Convolutional Networks have focused on optimizing the embeddings for each node via matrix-factorization-based objectives, without generalization of unseen data [5-8]. These algorithms are thus transductive, which means that all the nodes must be seen during the training time. The implication of this is that the graphs must be fixed and new nodes cannot be added after training.

In contrast, GraphSAGE (Hamilton et al, 2017) is a framework for inductive representation learning on large graphs [9], which means that the graph learning can generalize to "unseen nodes" (i.e. nodes not encountered during training time). Despite not having the full graph during training time, GraphSAGE did not compromise on accuracy and showed state-of-the-art results on datasets such as predicting Reddit communities [10].

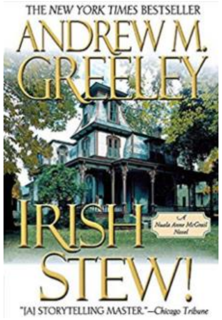
IV. Dataset and Features

The dataset we are using are Amazon items, extracted by SNAP, which consists of 548,552 products (393,561 books, 19,828 DVDs, 103,114 music CDs and 26,132 videos) [11]. After pre-processing the data, there are 127 different categories an item can belong to. On top of the meta-data (**Figure 1**), we also have the co-purchasing network, which outlines products that are frequently co-purchased with each other based on the "Customers who Bought this Item Also Bought…" feature on Amazon.



```
Id:   5
ASIN: 1577943082
  title: Prayers That Avail Much for Business: Executive
  group: Book
  salesrank: 455160
  similar: 5  157794349X  0892749504  1577941829  0892749563  1577946006
  categories: 2
   |Books[283155]|Subjects[1000]|Religion & Spirituality[22]|Christianity[12290]|Worship &
Devotion[12465]|Prayerbooks[12470]
   |Books[283155]|Subjects[1000]|Religion & Spirituality[22]|Christianity[12290]|Christian
Living[12333]|Business[297488]
  reviews: total: 0  downloaded: 0  avg rating: 0
```

Figure 1: Example metadata information


THE NEW YORK TIMES BESTSELLER
ANDREW M. GREELEY
A Nuala Anne McGrail Novel
IRISH STEW!
"[A] STORYTELLING MASTER."—Chicago Tribune

For our baseline, we used solely the title of the item to predict the category. A title often contains some information about its category. For example, it might be obvious to a human that "Patterns of Preaching: A Sermon Sampler" can be categorized as "Religion and Spirituality". Nevertheless, this is still a very difficult problem – without context, it's difficult to know "Irish Stew!" should be classified under "Literature and Fiction".

We thus tackled this problem by leveraging on the information given in the co-purchasing network and performing graph learning methods on it. The focus of our experiment is to find out how exploiting information about similar items can improve the accuracy by providing the needed context to classify such titles.

V. Methodology

The main inventive step in our model is to combine the NLP methods with Graph methods by using the hidden state after the RNN (Gated Recurrent Unit (GRU) or Long Short Term Memory (LSTM)) step from the Item Title as the node embedding for the GraphSAGE algorithm which is performed on the co-purchasing network (**Figure 2**).

Sentence Classification with NLP Techniques. The baseline that we will establish is using simply the title of the item to predict the category of the product. This corresponds to the top half of **Figure 2**. To do so, we utilize a multi-class text classification using Convolutional Neural Networks (CNNs), Recurrent Neural Networks (with either Gated Recurrent Units (GRU) or Long Short Term Memory (LSTM)) in TensorFlow.

CNNs have widely been used for sentence classifications where the temporal ordering of words does not matter [12]. In the Amazon dataset, "Patterns of Preaching: A Sermon Sampler" would have likely given the same product category as something like "Sampler Sermon: Preaching Patterns". Combining the output of the CNN (with 1 conv layer and 1 max-pooling layer) with the RNN (GRU/ LSTM) architecture represent the current cutting-edge techniques for text classification problems, and have shown state-of-the-art results in other text classification problems.

Our code for the CNN, RNN (GRU/LSTM) has largely been adapted from an open-source implementation applied to classifying crime descriptions in San Francisco into crime categories.[1]

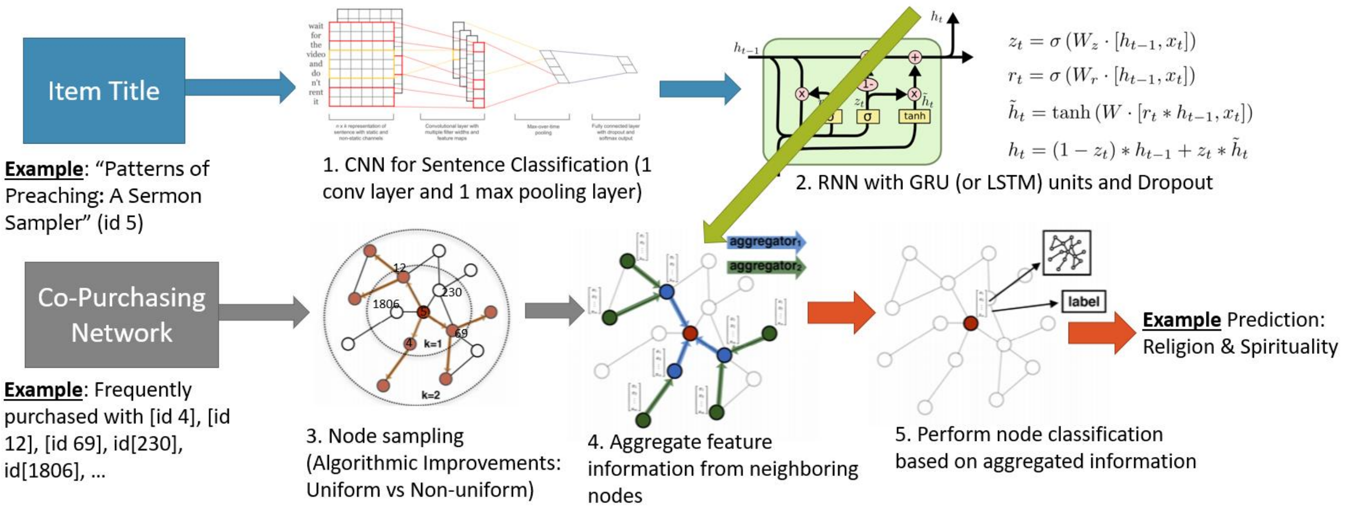

Item Title
Example: "Patterns of Preaching: A Sermon Sampler" (id 5)
wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output
1. CNN for Sentence Classification (1 conv layer and 1 max pooling layer)
h_{t-1}
h_t
z_t
$\tilde{h}_t$
1-
σ
tanh
$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$
$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$
$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$
$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$
2. RNN with GRU (or LSTM) units and Dropout
Co-Purchasing Network
Example: Frequently purchased with [id 4], [id 12], [id 69], id[230], id[1806], ...
12
1806
230
5
69
4
k=1
k=2
3. Node sampling (Algorithmic Improvements: Uniform vs Non-uniform)
aggregator₁
aggregator₂
4. Aggregate feature information from neighboring nodes
label
5. Perform node classification based on aggregated information
Example Prediction: Religion & Spirituality

Figure 2: Using the hidden state after the RNN (GRU/LSTM) as the node embedding for the GraphSAGE algorithm

Node Classification with GraphSAGE. The GraphSAGE algorithm corresponds to the bottom half of **Figure 2**. GraphSAGE works first by sampling a fixed number of nodes from its neighbors and aggregating feature information from these nodes. We used GraphSAGE largely due to its inductive representation, which allows us to generate node embeddings for unseen data. Our application to the Amazon dataset requires that we classify new items, which are "unseen nodes" in the graph. As such, a transductive method which requires all nodes in the graph to be present during training of the embeddings simply does not fit well with the evolving nature of the graph in the application area we are looking at.

The input embeddings that GraphSAGE currently uses for each node input feature are Word2Vec embeddings (averaged over the words if there are multiple words). However, we wish to experiment using the hidden states after the RNN (GRU/LSTM) as node embeddings instead (**Figure 2**), the intuition being that these hidden states encapsulate relevant information about the whole title.

Algorithmic Improvements on GraphSAGE. The current GraphSAGE algorithm uses uniform sampling, which randomly samples neighboring nodes to aggregate their feature information. This corresponds to the process represented by Point 3 in **Figure 2**. However, we posit that non-uniform sampling might work better if we can sample from nodes that give more valuable information.

We thus experiment with different sampling methods by leveraging on the graph information. In particular, we use the degree of the neighbors as a feature to select for when sampling. Our hypothesis is that sampling neighbors with lower degrees will give a higher accuracy since they are less likely to be "general" items that can be linked to many different classes of nodes. An example of a "general" item would be batteries, which are co-purchased with many possible items that belong to very different classes from razors to toys.

VI. Results

Combination of NLP and Graph Methods. We pre-processed our Amazon dataset and tested it using the sentence classification as well as the GraphSAGE algorithm. We took a 0.6 – 0.2 – 0.2 train-dev-test split for GraphSAGE, and achieved the following results (**Table 1**) on our models.

[1] Code largely adapted from: https://github.com/jiegzhan/multi-class-text-classification-cnn-rnn

Our baseline model of using NLP techniques on the Item Title achieves an accuracy of 0.233 and 0.267 (LSTM/GRU respectively), which is impressive given the difficulty of the problem. Exploiting the information in the co-purchasing network gives us an accuracy of 0.405 and 0.411 (LSTM/GRU), which is a 54% - 76% increase from our baseline of using CNN, RNN (GRU/LSTM) on Item Titles.

Table 1: Test accuracy of different methods

Methodology	Test Accuracy
Baseline: CNN, RNN (LSTM)	0.233
Baseline: CNN, RNN (GRU)	0.267
GraphSAGE (Uniform Sampling)	0.427
GraphSAGE (Non-Uniform Sampling)	0.428
CNN, RNN (LSTM) + GraphSAGE (Uniform Sampling)	0.405
CNN, RNN (GRU) + GraphSAGE (Uniform Sampling)	0.411
CNN, RNN (LSTM) + GraphSAGE (Non-Uniform Sampling)	0.409
CNN, RNN (GRU) + GraphSAGE (Non-Uniform Sampling)	0.410
Improvement over baseline	54% - 76%

Algorithmic Improvements for GraphSAGE. We tested our proposed algorithmic improvements on non-uniform sampling with the Reddit dataset, which was the dataset used in the original GraphSAGE paper. When testing the sampling methods, we used the mean aggregator as our aggregate architecture. Our results are reported in **Table 2**.

Table 2: F1 Score for Non-Uniform Sampling on Reddit Dataset

Sampling Methods	F1 Score
Uniform Sampling	0.947
Highest degree Sampling	0.919
Highest degree + Random	0.943
Lowest degree Sampling	0.946
Lowest degree + Random	0.950

After repeating further experiments with different random seeds, we found that choosing the lowest degree with randomness has a stable improvement compared with the uniform sampling. Non-uniform sampling has thus improved the accuracy over the reported baseline in the original paper.

Naïve Bayes. In addition to these methods, we decided to implement simpler algorithms taught in the class such as Naïve Bayes for comparison purposes. Surprisingly, the Naïve Bayes algorithm boasts an accuracy of 0.362 on the Amazon dataset (Top 5 Accuracy: 0.546), which outperforms some of the complex algorithms.

VII. Discussion

Our new model on Amazon Dataset. Incorporating the co-purchasing information adds a 54% - 76% increase in accuracy compared to our baseline, which confirms our hypothesis that the co-purchasing network indeed provides the necessary context for item titles.

However, there is little improvement in using the hidden states over Word2Vec as node embeddings. From **Table 1**, we note that the original GraphSAGE algorithm which uses Word2Vec works marginally better than our novel method of using the hidden states from the GRU/LSTM. This surprising result could potentially be due to the short titles of the Amazon items. We hypothesize that this method might prove more useful for document classification rather than for titles. Future work would include trying these techniques on other datasets to analyze if this method is suited for certain types of data.

Non-Uniform Sampling. As we have expected, the lowest-degree sampling has provided a slight improvement in accuracy on the baseline in the Reddit dataset. This confirms our hypothesis that nodes with lower degrees are less likely to be "general" items and hence provide more useful information. However, while this provided some improvements in the Reddit dataset, we found mixed results in the Amazon dataset. This technique has improved the accuracy marginally on the LSTM model, but not for the GRU model.

Our hypothesis for the lack of clear results on the Amazon dataset is that the distribution of degrees on the Amazon dataset has much less variance than the Reddit dataset. Without a huge variance in degrees, a lowest-degree sampling is approximately equal to a uniform sampling. We thus propose that our algorithmic improvements work best in datasets when there is a large variance in the distribution of node degrees.

Overall Accuracy. Overall, we have still not achieved extremely high levels of accuracy. Upon performing Error Analysis, we found that Amazon items often lend itself to multiple categories (out of 127 possible categories). For example, our NLP classification predicted "On Ethics and Economics" as "Social Sciences" when the ground truth label was "Reference". Further work would include cutting down to 20 general categories to tackle this "multiple classes" problem.

VIII. Conclusion

The focus of our project was to introduce a new way of synthesizing NLP techniques with state-of-the-art inductive graph techniques. This is a novel and unexplored approach as of yet, and future work would include experimenting with different NLP techniques (RCNN, Hierarchical Attention Networks) and other graph algorithms (Graph Attention Networks by Veličković et al [13]).

Nevertheless, we have shown that utilizing graph information often provides necessary context for difficult problems, such as category classification for Amazon items. Our current algorithm has shown improvements over the text-only baseline, and has the potential to suggest suitable categories for new items listed, saving time for sellers on Amazon.

IX. Contributions

Joseph has focused on the NLP methodologies for category classification of the titles (CNN, RNN (GRU and LSTM), Word Embeddings); Fengjiao has focused on data processing and testing GraphSAGE on the dataset; Yaqing has focused on testing algorithmic improvements to the GraphSAGE algorithm. Furthermore, we wish to thank William L. Hamilton for his advice and help on this project. We also thank the CS229 teaching team for their helpful comments and insights.

X. References

[1] W.L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. arXiv preprint, arXiv:1603.04467, 2017.

[2] S. Cao, W. Lu, and Q. Xu. Grarep: Learning graph representations with global structural information. In KDD, 2015.

[3] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In KDD, 2016.

[4] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In KDD, 2014.

[5] T. N. Kipf, and M. Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[6] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. In NIPS, 2001.

[7] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In WWW, 2015.

[8] D. Wang, P. Cui, and W. Zhu. Structural deep network embedding. In KDD, 2016.

[9] Jure Leskovec and Andrej Krevl (Jun 2014), SNAP Datasets: Stanford Large Network Dataset Collection, http://snap.stanford.edu/data

[10] W.L. Hamilton, R. Ying, and J. Leskovec. Representation Learning on Graphs: Methods and Applications. arXiv preprint, arXiv: 1709.05584, 2017.

[11] J. Leskovec, L. Adamic and B. Adamic. The Dynamics of Viral Marketing. ACM Transactions on the Web (ACM TWEB), 1(1), 2007.

[12] Y. Kim. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882. 2014

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. arXiv preprint arXiv:1710.10903, 2017.