Improving Product Categorization from Label Clustering

Alexander Friedman
Stanford University
ajfriedman@stanford.edu

Alexandra Porter
Stanford University
amporter@stanford.edu

Alexander Rickman
Stanford University
arickman@stanford.edu

ABSTRACT

From Amazon web-crawl data, we obtain a network of labels, where the weight of an edge between two nodes is determined by how many books have both labels. While the labels are organized into a hierarchy by Amazon, it contains numerous redundancies and uninteresting labels which reduce the useability as a user-facing shopping tool. To address this problem, we propose a method of exploring and visualizing the label graph so that a more effective organization can be implemented. We use node2vec [3] to compute feature representations of the nodes and then apply clustering to identify improvements that should be made to the labeling system. We concluding by discussing findings involving anomaly detection, identification of redundant or closely associated labels, and label hierarchical organization.

1 INTRODUCTION

1.1 Motivation

In a massive online store such as Amazon, keywords to describe books can easily be acquired by either seller input or automatic searching of the text. However, the size and organization of the set of labels can quickly become intractable, making it difficult to assign a clean and concise categorization hierarchy. Our goal is to determine how a set of labels applied to a set of books should be organized into a categorization system.

We implement an algorithmic and application based project to analyze data from Amazon web-crawl data of books and their categorizations. We interpret this as a network, in which category label strings are nodes. Edges of the network indicate which labels appear together: for each pair of labels shared by a book, we add an edge between the labels. Here we take a novel approach of applying clustering techniques to achieve our goal.

1.2 Related Work

1.2.1 Graph Clustering. Recent work on labeled graph clustering includes "Using Node Identifiers and Community Prior for Graph-Based Classification" [13], in which Ye et al. propose ways to implement traditional classification algorithms to make predictions as to the labels of nodes in a graph. They propose an algorithm called identifier-based relational neighbor classifier (IDRN) to solve the within-network multi-label classification problem. This paper provides clear motivation for traditional clustering on graph embeddings.

Zhou et al. [14] present another graph clustering method for complex networks, using a novel approach to node similarity based on attracting and recommending power of nodes.

1.2.2 Studying Online Stores. Our research was applied to Amazon categorizations of products. Other approaches to machine learning on product categorization include "Understanding How Product Attributes Influence Product Categorization: Development and Validation of Fuzzy Set-Based Measures of Gradedness in Product Categories" [7], which describes a way to apply fuzzy set theory to better model product categorizations that involve both discrete and continuous data. By evaluating set membership with degrees, Childers and Viswanatha were able to get more nuanced categorizations and evaluate membership over multiple categories. "The Future of Retailing" [2] surveys key areas of technology in stores, including visual displays and tools to facilitate decision making. In particular, they cite [4], which suggests that retailers need to make assortments easier for customers to understand, including by reducing the size of the selection. This concept applies to online stores, and is our motivation for this work.

Liu et al. [6] analyze the Google Play app market with a goal similar to ours: to determine the class hierarchy and unique relationships. However, they take the approach of crowd sourcing ground truth labels to combine with an NLP step of the app keywords to train a classifier.

1.2.3 Applications and Extensions of Node2Vec. Other works applying Node2Vec include "Node2vec in Telco: An Application of the Novel Feature Learning Method for Predictions in Call Networks" [8], which demonstrates how Node2Vec can be applied to a call network of customers of a telecommunication company in order to predict caller characteristics such as age and gender. They performed this under a semi-supervised learning regime, where a fraction of the customers provided information on these topics (known labels), with the goal of predicting the unknown labels for the remaining customers.

Extensions of Node2vec include metapath2vec [1], which incorporates metadata on different types of nodes and edges, and struc2vec [10], which learns node representations based on their structural identity in the graph.

2 METHODS

We analyze the relationships between labels of books using Node2Vec *node embeddings* and clustering methods. The Node2Vec algorithm generates real-valued feature vectors for each node in the graph for some selected dimension d. We then perform clustering on these points in d-dimensional space to determine groupings of nodes. The parameters of Node2Vec allow us to emphasize either the structural similarity or connectivity of nodes. Thus by analyzing multiple embeddings by applying clustering, we can learn which labels are similar in both role and actual meaning.

CS229, Fall 2018, Stanford University

2.1 Dataset

Table 1 summarizes the data from the Amazon web crawl, which was compiled in Summer 2006. The dataset contains Amazon products and the category groupings ("labels") to which they belong. To compile our graph dataset, we created a node for every label which a book belonged to, and created edges between two labels if a book belonged to both labels. The weight over edges corresponds to the number of books with both labels.

Labels in the original Amazon dataset can be described as a forest, and there are multiple trees to which a book may belong. For example, one book in the original dataset belongs to two trees with labels: 1.) **Books > Subjects > Arts & Photography > Photography > Photo Essays** AND 2.) **Amazon Web Store > Categories > Camera & Photo > Photography Books > Photo Essays**

These categories are somewhat redundant, and one of the goals of our model will be to detect categories which can be merged or used to provide additional recommendations to a user.

Number of books	393,561
Book categories	14,874
Most common categories	
Nonfiction	55,923
Children's Books	46,533
Religion and Spirituality	43,528
Literature and Fiction	41,899
Professional and Technical	41,838

Table 1: Dataset Statistics

2.2 Node2Vec

The main idea of Node2Vec is that we want to represent the vertices of the graph such that vertices "close" together have similar representations, where this closeness is some mixture of proximity in the graph and similarity in role, or neighborhood structure. The Node2Vec algorithms samples a set of random walks and then performs stochastic gradient descent on the feature representation of the vertices, where the loss function is the similarity of the pairs of representations given the vertices appear together.

We first describe how embedding is set up as a stochastic gradient descent method. Let $f : V \rightarrow \mathbb{R}^d$ be mapping to features representation; i.e. f is $|V| \times d$ parameter matrix. For $u \in V$, $N_S(u) \subset V$ is neighborhood with sampling strategy S. Maximize objective function: $\max_f \sum_{u \in V} \log Pr(N_S(u)|f(u))$. Conditional independence is assumed such that this becomes: $Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u))$. Since the network is un-directed, relationships between nodes are symmetric: $Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u)}{\sum_{v \in V} \exp(f(v) \cdot f(u))}$. Thus the maximum function simplifies to

$$\max_f \sum_{u \in V} \left[-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \right]$$

.

Node2Vec allows for random walks to be selected "between" Depth-First Search and Breadth-First Search strategies. This is accomplished by using parameters which weight the probability of a walk step returning to the source and the probability of exploring further into the graph. These parameters define the sampling strategy S used to generate the neighborhoods in the above formulas.

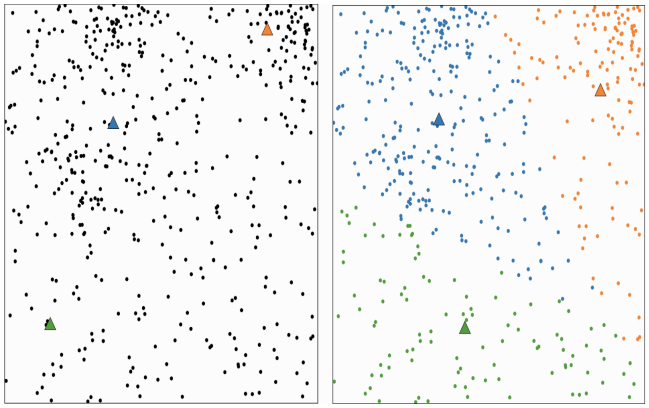


Figure 1: K-Means: Onset to Convergence [12]

2.3 Clustering Methods

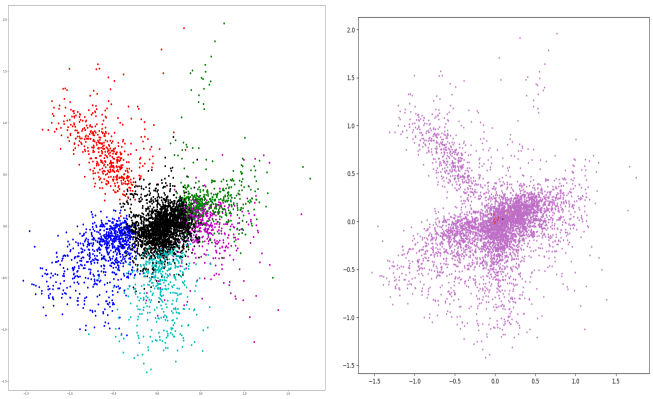
Above we discussed the motivation for an algorithm from which we use clustering techniques to improve the modularity of a given network based on co-purchasing data. In this section, we dive deeper into the clustering algorithms implemented. In each iteration, as discussed above, we use Node2Vec and then K-means to determine optimal modules. Before discussing the setup, in Algorithm 1 we provide the pseudo code for the K-means algorithm. K-means runs with O(n*k*t), where n is the number of iterations, k the cluster number, and t the number of data points. [5]

Algorithm 1 K-Means Algorithm

1: **procedure** K-MEANS(k)
2: Select k points at random as cluster centers
3: Assign objects to closest centroid by Euclidean distance
4: Calculate the centroid or mean of all objects in each cluster
5: Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds. [11]

Being that we cannot a priori estimate the number of product categories corresponding to the number of clusters, k, to set our algorithm searching for, we hypothesized that running an algorithm such as DBSCAN (density-based spatial clustering of applications with noise) on the data before could improve performance by finding the optimal clustering number for us. DBSCAN takes in inputs of the radius and minimum number of data points for a cluster, and determines the optimal number of clusters based on this. DBSCAN is robust to noisy data sets and would seemingly be efficient in discarding outliers in our Node2Vec represented network before our iteration segment of the algorithm designed to do this even begins. We ultimately decided to abandon this DBSCAN step in our code on the basis of its low performance compared to manual selection of cluster number for K-means. This project can be taken further by studying the reasons for this, and tuning the logic and/or parameters to reintegrate DBSCAN into this method more

constructively. Figure 2 shows the result of K-means and DBSCAN applied to the same data set (generated from our network Node2Vec representation discussed above), indicating our choice to abandon this approach.



((a)) ((b))

Figure 2: (a) K-Means clustering (6 clusters), (b) DBSCAN

2.4 Method Framework

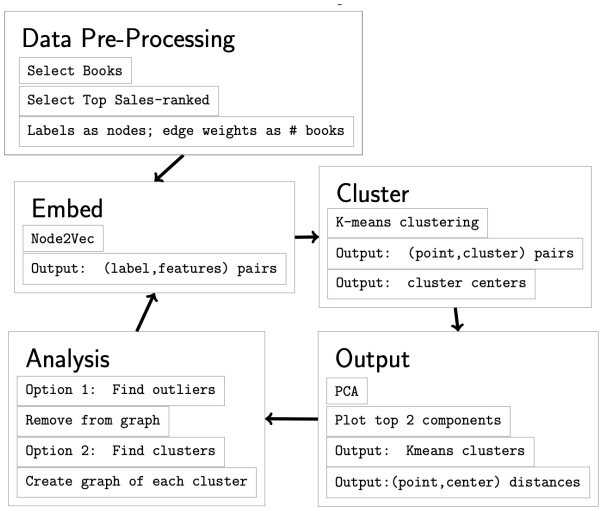


Figure 3: Full Method Workflow Diagram

As shown in Figure 3, after interpreting the web-crawl data as a network, we iterate through a workflow of embed, cluster, plot, analyze, and repeat. In this process we adjust parameters of both the Node2Vec and clustering models. We can use this system to detect/remove outliers before optionally re-embedding. We can also select a cluster from the initial run, then re-embed and re-cluster that cluster, repeating numerous times in order to collect redundant categories and analyze label hierarchies. After analysis, we select an induced subgraph of the original graph to re-embed and continue the cyclic process. We use the scikit-learn package to cluster and plot [9].

Throughout the process, the clusters of smallest radii indicate groups of labels which may be similar enough to combine into one category. Outliers indicate labels which are not closely related to any others; in practice these are labels which do not need to be included in a user-facing system or simply the lowest level (i.e. most specific) labels. Since we do not have a ground-truth into how labels should be interpreted, our tool is designed to present options for improving the label set to a user who would not be able to parse through the massive label set any other way.

Distance from Center	Label
1.9203707947953235	Subjects[1000]
1.9626659852879147	Instruction[11811]
2.0156069220765436	Books[283155]
2.0276880269223216	Poetry[9966]
2.1297687095091673	Foreign Languages[11773]
2.2177811099675195	Dictionaries & Thesauruses[11475]
2.4396388017039103	General[725800]

Table 2: Anomalous labels

2.5 Analysis Methods

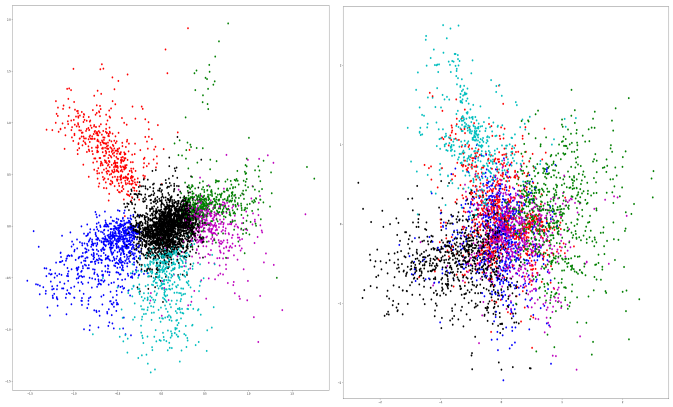
The main result of our system is a visualization of the label space. While we use the full dimensionality of the embedding to cluster and identify outliers numerically (by distance from cluster center) our system is also useful as a user-facing tool. We use PCA to select two dimensions for plotting. We compared this to 3-dimensional plots and plots of other dimensions besides the principal ones, but the 2-dimensional PCA plots conveyed the full structure of the point set while being much more concise.

3 EXPERIMENTS

We present visualizations of the label space created using two variations of our workflow (Figure 3); one in which we remove anomalies to determine if they had a disproportionate effect on the embedding, and one where we recursively repeat the process on each cluster to divide the data into a hierarchy.

3.1 Anomaly Detection and Removal

We use Euclidean distances of points from K-means centroids to detect outliers (as seen in the table below). We can directly remove these outliers from the plots, but we hypothesized that removing outliers from the graph and re-embedding before re-plotting would produce more cohesive clusters. As seen in Figure 4, removing anomalies results in less clearly defined clusters, likely due to the cluster structure being primarily defined by the anomalies. We hypothesize that the graph induced by non-anomalous nodes is relatively uniform and thus lacks structure for our method to identify. Table 2 lists the top anomalies removed, which are all fairly general labels, including some that may not be useful for a uer at all within a book store, such as “Books” and “General.”



((a)) ((b))

Figure 4: (a) Original clustering (6 clusters), (b) Anomalies removed from graph and re-embedded before another clustering.

Cluster 1	Cluster 2	Cluster 3
Regions[17228]	Computer	Arts
Regions[640504]	Computers	Camera
States[17263]	Design	Categories[493964]
States[640538]	Digital	Collections,
United	Internet[768564]	Collections,
United	Programming[3839]	General[2050]
	Project	Photo
	Software	Photo
	Specialty	Photographers,
	[229534]	Photography
		Photography[2020]
		[172282]

Table 3: Examples of clusters identified

3.2 Nested Label Associations

After two iterations of embedding and clustering, we see that groups are mostly made up of labels which are redundant or closely related. The first pass on embedding and clustering, show in Figure 5, is constructed using 10 random walks of length 10, with embedding parameters $p = 0.1, q = 1$, and clustered with $k = 4$. The outliers identified in blue boxes include Table 3 shows examples of label sets clustered together, as visualized in Figure 6; each subplot of Figure 6 is created by selected the subgraph of the origin network induced by the nodes of a cluster in Figure 5 and running the full embed and cluster process. Thus we can interpret these results as a two-layer hierarchy, where the clusters in Figure 5 specify the groups corresponding to nodes in the upper layer and the clusters in each plot of Figure 6 specify the children of each of these nodes. Note that in all plots, multiple labels with the same text appear; this means there are multiple underlying system tags corresponding to that label string.

As indicated by the annotations in Figure 6, we found the following groups of labels which are related and possibly redundant, especially because they also appear together as outliers relative to the main clusters. Some of these relationships, such as between "Professional & Technical" and "Medicine", or "Sacred Text" and "Bible", are not obvious from the labels themselves, but indicate that these labels are most often used together in this dataset.

- Europe (Fig. 6(a))
- Photography, Camera, Photo (Fig. 6(a))
- United States, Regions (Fig. 6(a))
- Bible, Sacred Text, Christianity (Fig. 6(b))
- Mystery, Suspense, Thrillers (Fig. 6(b))
- Professional Science, Medicine, Professional & Technical (Fig. 6(c))
- Computer & Internet Books, Software Design, Specialty Stores, Digital Business & Culture, Design, Development, Project Management (Fig. 6(d))
- Pure Mathematics, Applied, Physics, Sciences, Mathematics, Engineering (Fig. 6(d))

The following label sets appear as anomalies. Labels called "General" most often appear as leaves in the label tree in the Amazon labeling system and thus appear in a wide variety of categories.

- Guidebook, Guidebook series (Fig. 6(a))
- Accessories, note cards (Fig. 6(a))
- Authors & Illustrators, A-Z, General, Ages 9-12, History & Historical Fiction (Fig. 6(b))
- Medical (Fig. 6(c))
- Science (Fig. 6(c))
- Books, Subjects, Entertainment, General, Education (Fig. 6(c))
- Amazon.com Stores, General (Fig. 6(d))

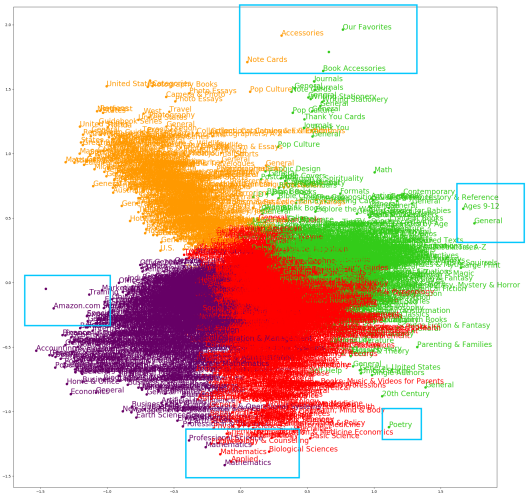


Figure 5: Embedding and clustering of top sales-ranked books.

4 CONCLUSION & FUTURE WORK

We have shown that our method produces a visualization tool for understanding the label set, including anomalous and redundant

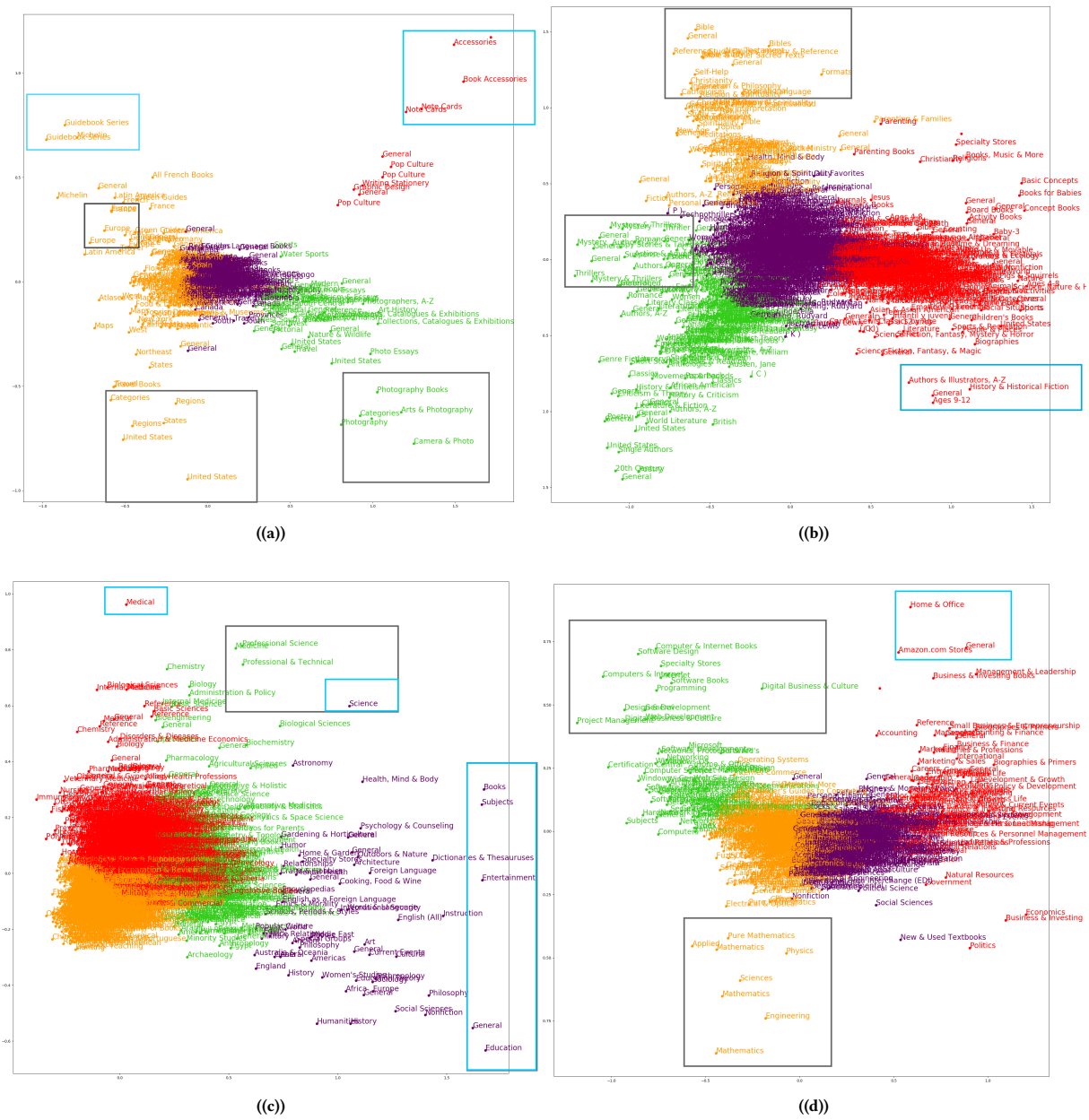


Figure 6: Annotated results of the embed and cluster process applied to the clusters in Figure 5: (Figure 5 cluster, Figure 6 subplot) relationships are (Orange, Figure 6(a)), (Red, Figure 6(b)), (Purple, Figure 6(c)), (Green, Figure 6(d))

labels. While mostly accurate, some labels did appear in unexpected clusters, most likely due to books which fit multiple categories and thus add edges between very different labels.

The next step that should be taken with these results is to establish a ground truth based on crowd-sourced human preferences for labels, since the end-goal is human readability. Further research could also examine optimization of node2vec parameters, including search strategy. Parameter tuning could also be applied to K-Means clustering and outlier thresholds. Additionally, further research could look at the necessary number of nested label clustering steps and re-embedding steps to find all redundancy. And finally, similar methods could be applied to financial transaction networks, telecommunication networks, and healthcare data.

REFERENCES

[1] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 135–144. ACM, 2017.

[2] D. Grewal, A. L. Roggeveen, and J. Nordfält. The future of retailing. *Journal of Retailing*, 93(1):1–6, 2017.

[3] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[4] B. E. Kahn. Using visual design to improve customer perceptions of online assortments. *Journal of retailing*, 93(1):29–42, 2017.

[5] Kldavenport.com. The cost function of k-means, 2018.

[6] X. Liu, H. H. Song, M. Baldi, and P.-N. Tan. Macro-scale mobile app market analysis using customized hierarchical categorization. In *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2016.

[7] T. L. C. Madhubalan Viswanatha. Understanding how product attributes influence product categorization: Development and validation of fuzzy set-based measures of gradedness in product categories. *Journal of Marketing Research*, 1999.

[8] B. B. MarÃŋa ÃŞskarsdÃştir. Node2vec in telco: An application of the novel feature learning method for predictions in call networks. *DataMiningApps*, 2016.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394. ACM, 2017.

[11] Saedsayad.com. K-means clustering, 2018.

[12] S. University. Visualizing k-means clustering, 2018.

[13] Q. Ye, C. Zhu, G. Li, Z. Liu, and F. Wang. Using node identifiers and community prior for graph-based classification. *Data Science and Engineering*, 3(1):68–83, 2018.

[14] H. Zhou, J. Li, J. Li, F. Zhang, and Y. Cui. A graph clustering method for community detection in complex networks. *Physica A: Statistical Mechanics and Its Applications*, 469:551–562, 2017.

A APPENDIX: CODE

Code can be found at: https://github.com/aporter468/embedandcluster
Note that the necessary Node2Vec library is not included, it can be found at: https://github.com/aditya-grover/node2vec/tree/master/src

B APPENDIX: CONTRIBUTIONS

Friedman: Experimented with different combinations of DBScan, K-Means, PCA, and colors to create visualizations. Helped write motivation section and ran preliminary data characterization to help create graph interpretation of dataset.

Porter: Implemented code for converting web crawl data into a graph, selecting subgraphs induced by sets of labels, computing cluster distances, and constructing label lists for plots. Ran node2vec embeddings and analyzed results. Performed literature review.

Rickman: Researched and tested different clustering algorithms and devised a scheme to apply DBSCAN and K-means in sequence as a potential means to achieve our categorization goal described above. Optimized clustering parameters to improve performance and visualization.