Improving Product Categorization from Label Clustering

Alexander Friedman, Alexandra Porter, Alexander Rickman

CS229: Machine Learning Class Project

{ajfriedman, amporter, arickman}@stanford.edu

Motivation

In a massive online store, an intractably large set of keywords to describe books can easily be acquired by either seller input or automatic searching of the text. Our goal is to organize a massive set of labels applied to a set of books to use for categorization. We implement an algorithmic and application based project to analyze data from Amazon web-crawl data of books and their categorizations. We embed labels into a feature space, and apply clustering approaches to find interesting features such as redundancies, hierarchies, and anomalies.

Methods

- **Node2vec** The node2vec algorithm [1] samples a set of random walks and then performs stochastic gradient descent on the feature representation of the vertices. The loss function is the similarity of the pairs of representations, given that the vertices appear together.
- **Clustering** Once we have node2vec representations of the network, we cluster with K-means [3]. Based on subjective observation and testing on the data set, we specified the number of clusters as 6:
  1. procedure K-MEANS(k, pointset)
  2. while centers change do
  3.   cluster centers = k random points
  4.   for c ∈ cluster centers do
  5.     center[c] = argmin Dist(p, c)
  6.   for c ∈ cluster centers do
  7.     c = mean({p | center[p] = c})

Dataset

The Amazon dataset contains metadata on 350,000 books, including the categories ("labels") to which each book belongs. The graph dataset which we input into Node2vec was created by using labels as nodes and generating edges between nodes whenever a book belonged to multiple labels. Labels in the original Amazon dataset can be described as a forest. These labels can often be redundant, so our model aims to detect these redundancies so they can be replaced with a cleaner labeling scheme.

Framework and Implementation

- **Data Pre-Processing**
  - Select Book
  - Select Top Bader-ranked labels on nodes, edge weights on # books

- **Embed**
  - Input: labels, feature pairs

- **Analysis**
  - Option 1: Find outliers
  - Remove from graph
  - Create subgraphs

- **Cluster**
  - K-means clustering
  - Input: (point, cluster) pairs
  - Output: cluster centers

- **Output**
  - Embed and cluster (6 clusters)
  - Initialize by nodes in magenta cluster

Results

- **Anomaly Detection and Removal**

- **Nested Label Associations**

![Figure 2: (a) Original clustering (6 clusters), (b) Anomalies removed from graph and re-embedded before another clustering.](image)

![Figure 3: Creating nested clusters.](image)

Analysis

- **Anomalies** 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 2(b), 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.

![Distance from Center (label)](image)

Dataset

The Amazon dataset contains metadata on 350,000 books, including the categories ("labels") to which each book belongs. The graph dataset which we input into Node2vec was created by using labels as nodes and generating edges between nodes whenever a book belonged to multiple labels. Labels in the original Amazon dataset can be described as a forest. These labels can often be redundant, so our model aims to detect these redundancies so they can be replaced with a cleaner labeling scheme.

References


Future Work

- Additional parameter optimization: node2vec search strategies (depth vs. breadth), k-means clusters, outlier threshold.
- Determine necessary number of nested label clustering steps to find all redundancies.
- Additional applications: other product categorizations, financial transaction networks, telecommunications networks, pharmaceutical co-prescription data.