Clustering for Taxonomy Evolution

Sneha Bankar and Anindya Das
Computer Science Department
Iowa State University

Abstract

Products on shopping sites like Amazon and eBay follow a taxonomy and are placed in their respective categories. Machine learning classifiers are trained using product descriptions and the products are placed in the categories. However due to addition of new items day to day to the product catalogue, it may be necessary to update the taxonomy. We propose using clustering techniques to detect taxonomy evolution. Our work is mainly focused on analyzing clustering algorithms to cluster the products into categories based on product descriptions.

Introduction

In various shopping portals like eBay or Amazon, products are categorized in a catalogue and they appear in the form of a tree. When a customer wants to browse for a particular product, s/he can traverse the tree structure until s/he reaches the desired product category. For example if a customer is looking for a Digital SLR Camera he will browse through the following tree structure:

- Camera and Photo
  - Lenses
  - Flashes
  - Digital Cameras
    * Compact System Camera
    * Digital SLR Camera
    * Point and Shoot Camera

Companies like Amazon use machine learning techniques to place products for sale into these categories. From the above example, we can see Digital Cameras can further be divided into 3 product sub-categories. However with time there could be a case where a product from a new sub-category is coming up but due to lack of a sub-category they are being placed into one of the given 3 sub-categories. We call this case as Taxonomy Evolution. There is no way to determine if the taxonomy is evolving without human supervision. We propose use of Clustering mechanism to detect taxonomy evolution in an unsupervised way. We could take products from all the sub-categories together and apply Clustering Algorithms to categorize them. After clustering if we find that more sub-categories are derived, it means our taxonomy has evolved and there is a need of introducing a new sub-category.

Taxonomy evolution is a rather new problem and clustering approach has not yet been used to address this problem. The clustering problem is defined as to find groups of similar objects in the data. The similarity between the products is measured with the use of a similarity function. Each product is represented as a vector based on presence or absence of a word in a product description. We have worked on a connectivity based clustering algorithm called as Hierarchical Agglomerative Clustering and a centroid based clustering algorithm called as K-Means. After forming clusters of products, the most important thing is evaluation of clusters before deciding on taxonomy evolution. We use a new approach of cluster evaluation which could be a good fit for analyzing text clusters. Our project mainly aims at forming good clusters out of products and if good quality clusters are formed then decides on taxonomy evolution.

Data Collection

Since we require data which consists of products placed in different categories we used the Amazon Product Advertising API to get the product data. From the browse tree of Amazon we selected few leaf nodes and applied clustering techniques to these leaf nodes. Each leaf node represents a category.

Camera Data Corpus- This contains data from 10 different categories like Compact System Cameras, Compact System Camera Bundles, Digital SLR Cameras, Digital SLR Camera Bundles, Point and Shoot Digital Cameras, Point and Shoot Digital Camera Bundles, Medium Format Digital Cameras, Camcorder Lenses, Camera Lenses, Underwater Lenses. All these nodes altogether have 1790 items and the number of non-unique features in the product descriptions 9450.

Coffee-Tea Data Corpus- This contains data from 17 different categories like Whole Bean Coffee, Ground, Coffee Pods, K-Cup Coffee, Instant, Capsules, T-Discs, Unroasted...
Beans, Coffee Gifts, Black tea, Chai, Green tea, Herbal tea, Oolong tea, Tea gifts, Tea Samples. All these nodes altogether have 800 items and the number of non-unique features in the product descriptions 4300.

**Feature Extraction**

Here, we have collected the raw description of each item. Description is a list of words, where each word can be considered as a feature. This raw description is not good to use for clustering. We need to extract features in such a way that they talk about the item. Features should not include anything which is redundant. The techniques we have used are described below:

**Removal of Punctuation:** We removed all punctuation marks added with the words in the description. The set of forbidden characters includes all punctuation marks and brackets. We excluded '-' because it is used as a connector of words which may be necessary to express a good feature.

**Removal of Brands:** For some sets of data, we had brand information available. For example, in case of data of Camera, the description can contain 'Canon' as the manufacturer of the product. Now, Canon makes all sorts of camera, so it will be present in the description of all types of cameras. So it does not give us any information about the classification of cameras. Moreover, it will have a very high frequency, which may pollute the distance measure. Therefore, we remove this information from item description.

**Removal of Stop words:** Stop words are words which we can exclude from any description. Articles (a, an, the), Prepositions (in, for, from, with, within etc.) and conjunctions (and, or, but etc.) are included in this list. These words don’t contain information about the items.

**Stemming:** Stemming is used to reduce the words to its ‘root’ or ‘stem’ form. For example, *include, includes, including* - all these words can be reduced to the root form *include*. This is useful, because in this way, we can say that the same feature is present in all 3 descriptions, which will help to determine the similarity and distance of items.

**Unigram and Bigram:** After applying all the techniques described above, each of the remaining modified words is considered as a Unigram. Bigram is the conjunction of two successive words in the description. For example, *smartphone and attachable* are two Unigrams. If we consider them only as separate features, then we ignore the information that actually this item is attachable to smartphone. Another item description which also contains these two words, but not as two consecutive words, would convey the same information. Therefore, we construct a Bigram *smartphone attachable* to incorporate this characteristics of the item.

**Feature Frequency:** Feature Frequency is the number of items who have this feature. This is used as weights for both clustering methods. Sometimes we have considered only top 5 – 10 frequent features to compute distance.

**Distance Measures**

In this section, we are going to discuss 4 different distance measures which we have used here. These measures are used to determine the distance between two items. The following measures are used:

1. **Edit Distance:** This distance is defined by the minimum no. of insert, delete or replace operations necessary to convert one description to another. Here, we have used the weighted version, i.e. the distance is the summation of weights of the features involved in those insert, delete or replace operations.

2. **Euclidean Distance:** This distance is defined by the sum of the square of the weights of disjoint features between two descriptions. Here, disjoint features mean the features which are present in only one of them, not in both.

3. **Jaccard Distance:** Jaccard Similarity is defined by the ratio of the intersection and union of two sets. Here we have defined it similarly. Then we subtract it from 1 and that gives us the Jaccard distance.

4. **Hamming Distance:** This is the number of disjoint features between two descriptions.

**Clustering Techniques**

We have used two clustering algorithms: Hierarchical Agglomerative Clustering (HAC) and K-means Clustering. Both of these methods are frequently used to solve different types of clustering problems, but we needed to tweak the algorithms a little bit. The algorithms are described below:

**Hierarchical Agglomerative Clustering**

In this algorithm, initially all the items are considered as distinct clusters. So, in case of n products, we have n clusters at the beginning. Then the closest pair among all clusters is chosen. Then these two clusters are merged into a new cluster. This new cluster is inserted into the set of all clusters and the two clusters are deleted from the set. This is continued till the terminating condition satisfies. Here, we have defined the terminating condition with the number of clusters in the current set. If that is equal to the specified number of clusters, then we output these clusters. The Pseudocode is given below:
The distance between two clusters is measured by Unweighted Pair Group Method with Arithmetic mean (UPGMA). Say, we want to measure the distance between cluster $A$ and $B$. It is defined by the following equation:

$$UPGMA(A, B) = \frac{1}{|A| \times |B|} \sum_{x \in A, y \in B} d(x, y)$$

Here, $d(x, y)$ is the distance between the two items $x$ and $y$, where we can use any distance measure described earlier. We have used both edit distance and Euclidean distance.

**K-Means Clustering**

In this method, $K$ points are randomly chosen as centroid of $K$ clusters. Then each item is assigned to the cluster whose centroid is the closest to it among $K$ clusters. Then the centroid of each cluster is updated. The item from which the sum of distance to all other items of that cluster is minimized is assigned as the new centroid. Then we continue the iteration till the centroids don’t change. We have used both Euclidean distance and Jaccard distance in this algorithm.

Here, we have considered only top 5 frequent features for both items to calculate the distance, because less frequent features makes the distance measures confusing.

**Cluster Evaluation**

The main challenge in text clustering is evaluating if the clusters formed are good or not. Many a times, the goal of clustering could drive us to obtain a good cluster evaluation methods. Our goal of clustering is to group products based on what they are talking about. For example we expect all products which contain “teabags” in the description fall into one group and all talking about coffee gifts fall together into another.

A reasonable assumption is that the top most frequent features appearing in a cluster tell us about what the cluster is about. Thus to compare two clusters, cluster 1 and cluster 2 we count the number of items in cluster 2 which are talking about the top most features in cluster 1. The definitions of recall and precision for each cluster are given below:

$$X = \text{no. of items having features in cluster } i \text{ from cluster } i$$

$$Y = \text{no. of items having features in cluster } i \text{ from other clusters}$$

$$Z = \text{number of total items in cluster } i$$

$$Recall \text{ for cluster } i = \frac{X}{Z}$$

$$Precision \text{ for cluster } i = \frac{X}{X+Y}$$

**Results**

The precision results for our two approaches work are listed in Table 1. Dataset1 refers to Camera Data Corpus, Dataset2 refers to Coffee-Tea Data Corpus and Dataset3 is a combination of both. The average precision value means the average of the precision values of all clusters.

Though the precision values are good, recall for these methods is between 20% to 30%. We have tried different distance measures for both techniques and found out that Euclidean distance works better for HAC than other distance measures and Jaccard distance works better for K-means than other measures. We have tried different values of $K$ for K-means starting from the number of distinct browseNodeIds. The best precision value is reported here.

**Conclusion**

We have tried to apply both a partitioned algorithm i.e K-Means and hierarchical clustering approach using different feature weighing mechanisms for clustering products. We find that recall for both the approaches is very low. Our method was tried on different data sets which had diverse homogeneity. We observe that there is a scope for increasing the recall. The recall is low due to placement of items of same category in different clusters.

With regards to distance measures, edit distance is not a good measure in this case because using the features as bag of word is better than rather considering the order of words appearing in the product descriptions. Using K-Means along
with Euclidean Distance is not advisable now because with a high dimensional feature space assigning data points to centroids become difficult. Jaccard Similarity works well in this case.

In future the recall could be improved by detecting outliers from different categories and then trying to merge the outliers to form the new clusters. Also the concept of topic modeling could be used to mine topics from product descriptions and clustering the products based on these topics. One more approach to do this would be to perform associative rule mining and mine rules out of cluster and then evaluate the clusters. For this case, while applying K-Means the initial seeds could be preselected since we already know the number of categories and some items present in those categories.

References


