Lab 06: Exploratory Data Analysis (EDA) with Clustering
CS3300 Data Science

Learning Outcomes

1. Apply clustering algorithms to unlabeled data
2. Characterize similarities and differences between clusters
3. Identify important features using statistical tests

Overview

In the first half of the class, we've been focusing on exploratory data analysis (EDA). You've primarily been using data sets with only a handful of variables. You could analyze each variable with visualizations and statistics to find relationships. High dimensional data sets, however, have too many variables for you to analyze each variable individually. We need to turn to more sophisticated techniques such dimensionality reduction and clustering.

In this lab, you are going to analyze 63,542 emails. You will convert the raw text into a feature matrix using a "bag of words" model. Each column of the feature matrix corresponds to one word, each row corresponds to one email, and the entry stores the number of times that word was found in that email. You will perform dimensionality reduction using the Truncated SVD method, cluster the emails, and compare the "inherent" structure to the given class labels.

Instructions

Part I: Load and Transform the Data

a. Repeat your process from Lab 5 to get a feature matrix and a SVD projection matrix for the emails. When using the `CountVectorizer`, use the parameters `binary=True` to generate a matrix of absence (0) / presence (1) values and `min_df=10` to exclude any words that do not appear in at least 10 emails.

Part II: Cluster the Emails

To perform further analysis, we want to cluster the emails. By clustering the emails, each message will be assigned a cluster id (e.g., 0, 1, 2, etc.).

a. Review the clustering algorithms available in Scikit Learn. Choose a clustering algorithm that you think would be appropriate for this data set.

b. Cluster the samples using the two SVD components you chose in Lab 5. The clustering algorithm should return a 1D Numpy array of cluster labels (e.g., 0, 1, 2, etc.) for each point.

c. Create a scatter plot along the 2 SVD components. Color the points according to their cluster labels.

d. Assess whether the cluster assignments. Visually, the data should form two distinct groups or clusters. The clustering algorithm should label the points so that all points in the same cluster have the same cluster id. (With two clusters, there should only be two cluster ids.) If the clustering does not look correct, try adjusting the parameters of the clustering algorithm you chose or choose a different algorithm.

e. Calculate a confusion matrix for the ham / spam labels versus the cluster labels.

Part III: Calculating Document Frequencies of Words

From part II, we now have clusters, but we don't know why certain those specific emails were clustered together.

a. Create a separate matrix for each cluster containing the rows for the points in that cluster.

b. Convert the matrices to CSC format. We will be accessing the data column-wise. Column indexing is significantly faster for the CSC format than the CSR format. (Hint: Review the `scipy.sparse` module.)

c. Calculate the document frequency of each word for each cluster. The document frequency is the number of documents that contain each word. Since the feature matrix is binary, we can simply sum along the columns. (Note: The resulting matrices should have the same shapes with one entry for each word in the vocabulary. If not, then you may have done the sum along the rows instead of the columns.)

d. What are the document frequencies of the words "love", "works", and "different" for the emails in each cluster?

Part IV: Find Enriched Words with Statistical Testing

We are going to use a Binomial test to determine if the number of occurrences of a given word in a given cluster is higher than what would be expected from the other cluster.

Here is the code for calculating the expected probability of a word based on cluster 1 and using it to test if the word is enriched in cluster 0:

```
cluster_1_expected_prob = doc_freq_cluster_1 / num_emails_cluster_1
pvalue = stats.binom_test(doc_freq_cluster_0, num_emails_cluster_0,
                          cluster_1_expected_prob, alternative="greater")
```

The null hypothesis that the relative document frequencies of the observed cluster are less or equal to those of the tested. The alternative hypothesis is that the document frequency is higher in cluster 0 than cluster 1.

a. Try testing if the words "works" and "love" are enriched in cluster 0.

For "works", you should get an expected probability of 0.109 for cluster 1 and observed probability of 0.041 for cluster 0. The Binomial test should return a p-value of 0.999, indicating that the observed frequency for cluster 0 is NOT greater than the frequency for cluster 1.

For "love", you should get an expected probability of 0.004 for cluster 1 and observed probability of 0.035 for cluster 0. The Binomial test should return a p-value of 0.0, indicating that the observed frequency for cluster 0 is greater than the frequency for cluster 1.

b. Wrap your code for part (a) in a loop to find enriched words for cluster 0. Calculate the p-value for every word. If the p-value < 0.05, add a tuple of (pvalue, word, cluster 0 document frequency to a list. (Hint: Iterate over the vocabulary_ dict to get each word and its index.)

c. Filter out any words that contain non-alphabetic characters. (Hint: Use the isalpha() method of strings.)

d. Sort the words in ascending order by their *p*-values and print out the first 200 words.

e. Repeat with the clusters reversed so you can find words enriched in cluster 1.

Reflection Questions

a. Make a guess as to why the emails might form two distinct clusters.

b. Compare the ham/spam labels to the cluster labels using the confusion matrix you generated. Are spam messages in both clusters or a single cluster? Are all of the messages in the clusters with spam labeled as spam?

c. Skim through the top 200 words for each cluster. Can you identify any patterns for either of the clusters?

d. Select the rows in the DataFrames for the emails in cluster 0. Print the top 25. Do the same for cluster 1. Do you the to and from addresses and subject lines provide additional help in identifying patterns?

e. The clusters represent email from two separate mailing lists. One mailing list is for the R programming language, while the other mailing list is for a university. Which mailing list contained all of the spam?

Submission Instructions

Save the Jupyter notebook as a PDF and upload that file through Canvas.

Rubric

Followed submission instructions	5%
Formatting: Report is polished and clean. No unnecessary code. Section headers are used. Plots are described and interpreted using text. Axes must be labeled appropriately. The report contains an introduction and conclusion.	5%
Part I: Load and Transform the Data	15%
Part II: Clustering the Emails	15%
Part III: Calculate the Document Frequency of the Words	20%
Part IV: Find Enriched Words with Statistical Testing	20%
Reflection Questions	15%
Exceeded Expectations	5%