

Article Classification with Centroid Classification

Group 1

June 19, 2024



- 1 Project Overview
- 2 Dataset and Pre-liminary Analysis
- 3 Data Preprocessing
- 4 Centroid/Variance Calculation



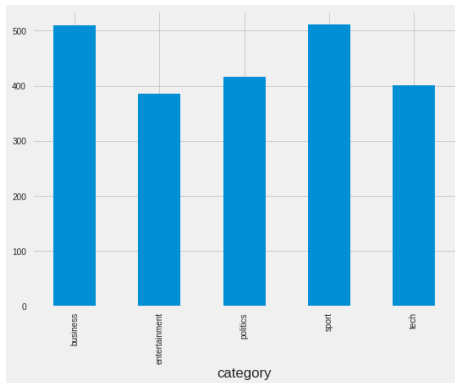
Project Objectives

- The goal of this project is classify document per category on given dataset.
- Document Embedding Model: Tf-Idf Vectorizer
- Classification Model: Centroid Classification



Dataset and Pre-liminary Analysis

- BBC ML Dataset
- Consist of 2225 articles, with categories: 'business', 'entertainment', 'politics', 'sport', 'tech'
- Dataset was changed from Wikipedia to BBC dataset due to complication in model training and dataset processing.



Data Preprocessing

- Category-Token Chi2 test

Category	1st	2nd	3rd
Business	oil	bank	growth
Entertainment	award	singer	awards
Politics	tory	blair	party
Sport	win	injury	coach
Tech	digital	computer	software



Tf-Idf Embedding

- Instead of plain BoW encoding, we use tf-idf scoring to encode the documents.
- The token value is calculated as follows:

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

$$idf(t) = \log \left(\frac{N}{n_t} \right) \quad (2)$$

$$tfidf(t, d) = tf(t, d) \times idf(t) \quad (3)$$



Tf-Idf Embedding

- Prior to embedding, unique words(unigrams and bigrams) are extracted from the dataset to be used as tokenizer.
- The tokens are filtered based on $\sum_{d \in D} tfidf(t, d)$, to filter out tokens with low occurrence, in our project it's set to 5. Resulting token size is 17989.
- For each document, the token value(vector index) in the vector is given by tf-idf scoring of document.



Document Vectorization and Clustering

- With the documents embedded, we now calculate the centroid and variance of each categories.
- Centroid is the main metric for the cluster, and variance is calculated to use gaussian distribution for probability distribution.
- No dimension reduction is applied to minimize information loss.
- We evaluate the model on normalized dataset and non-normalized dataset. For normalized dataset, we use distance metric $\frac{1}{\cos(\theta)}$, where $\cos(\theta)$ is the cosine similarity, and for non-normalized dataset, we use euclidean distance.



Centroid Classification

- Centroid classification is a simple classification method that uses the centroid and distance metric to classify the document.
- Our project uses 3 type of Centroid Classification: Nearest Centroid, Gaussian Probability Distribution, and Logistic Probability Distribution.
- The category prediction is generated by calculating highest score of probability distribution, or lowest distance from the centroid.



Nearest Centroid

- The simplest centroid classification method, where the document is classified based on the nearest centroid.
- The distance metric is calculated using the distance metric, and the document is classified based on the nearest centroid.
- The centroid is calculated as the mean of the document vectors in the cluster.



Gaussian Probability Distribution

- For each unique category, we construct a probability distribution using the centroid of the cluster and distance metric.
- Assuming central-limit theorem, we use a normal distribution to generate the probability distribution.

$$p(d) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d)^2}{2\sigma^2}}$$

Where d is the distance from the centroid, and σ is the variance of the cluster.



Logistic Probability Distribution

- Similar to Gaussian Probability Distribution, but instead of using normal distribution, we use logistic distribution.

$$p(d) = \frac{1}{1 + e^{-\alpha d}}$$

Where d is the distance from the centroid. We use α as a hyperparameter to tune the probability distribution, and it's set to 0.5.

- Logistic is used to generate a more smooth probability distribution, and is more robust to outliers.



Multi-Category Classification

- For multi-category classification, we use the probability distribution to predict the category of the document.
- We can tune the probability threshold for accuracy using the cost function:

$$J(C_d, C'_d) = \frac{\lambda_1 |C'_d \setminus C_d| + \lambda_2 |C_d \setminus C'_d|}{|C_d \cup C'_d|}$$

where C_d is document categories set, and C'_d is predicted categories set.

- Starting from threshold 0, we increase the threshold by a learning rate value until the cost function is minimized.
- We generate a candidate category for both pmf value using trained threshold, and we simply choose second closest centroid for Closest Centroid.



Evaluation

