Genre Classification of Songs Based on Lyrical Features

Tasneem Abed

University of the Witwatersrand

May 24, 2019

COMS4030A ACML Class Project 2019

Introduction

This project aims to implement 2 multiclass classification algorithms to try and predict the genre of a song based on its lyrical features:

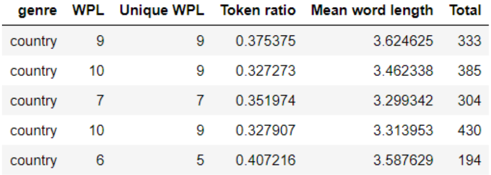
- K Nearest Neighbour using features extracted from the lyrics such as total number of words
- Naïve Bayes model using the bag-of-words representation

Data

- Scraped from the web using the Genius API
- Total of 3000 songs, 30 artists from each genre and 25 songs per artist
- 4 Genres: Country, Pop, Rap, Rock

Pre-processing

- Cleaned lyrics by removing punctuation, cue tags, removing duplicate lyrics
- Manually extracted the features in the figure below



genre	WPL	Unique WPL	Token ratio	Mean word length	Total
country	9	9	0.375375	3.624625	333
country	10	9	0.327273	3.462338	385
country	7	7	0.351974	3.299342	304
country	10	9	0.327907	3.313953	430
country	6	5	0.407216	3.587629	194

Figure: Image showing screenshot of data

Pre-processing - cont.

- The features have different ranges and the Total column's range is very large, normalize the data using rescaling:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

  Now all the data ranges between 0 and 1 to avoid bias.

Do these features contain enough information to indicate the genre of the song? If so, which ones have more influence? [Walsten and Orth]

Algorithms

- K Nearest Neighbour
  - Simple but effective algorithm that relies on a distance metric between data points in the feature space
  - Assigns the modal class of K nearest neighbours to a query datapoint

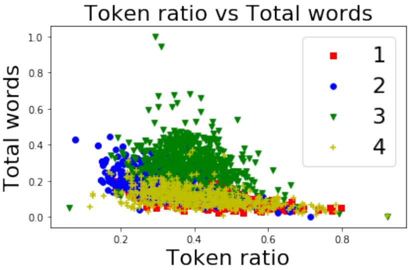


Figure: Plot of 2 features against each other

Algorithms - cont.

- Naïve Bayes
  - Conditional probability approach
  - Applies conditional independence assumption: $P(x|y) = \prod_{i=1}^{n} P(x_i|y)$ to Bayes rule

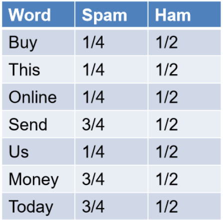
$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

giving the equation

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X) = \sum_{Y'} P(X|Y')P(Y')}$$

Bag-of-words

- The Naïve Bayes model uses the bag-of-words representation of text.
- There is a dictionary for each class, each containing every word found in the training data with the probability of the word appearing in that class
- Example



Word	Spam	Ham
Buy	1/4	1/2
This	1/4	1/2
Online	1/4	1/2
Send	3/4	1/2
Us	1/4	1/2
Money	3/4	1/2
Today	3/4	1/2

Figure: Bag-of-words example; Image taken from COMS3007 notes

Implementation

- Both algorithms implemented from scratch
- 20-80 test-train split
- Balance of classes
- Performance measured by accuracy of prediction i.e. how many predictions were correct over testing size and
- Classification error

$$\frac{1}{N}\sum_{n=1}^{N}[y^{(n)} \neq h(x^{(n)})]$$

where $h(x^{(n)})$ is the predicted class.

Initial Results

Naïve Bayes algorithm was slow and therefore tested once on the full dataset as well as on a subset of it.

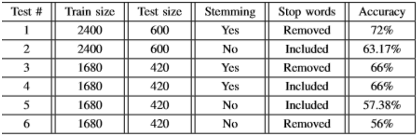
Achieved accuracies of 63.17% on the full set and 57.38% on the smaller set.

2 improvement techniques were applied: Stemming and stop words removal

Techniques

- Stemming - A process to reduce words down to its root (stem) form. Eg: plays, played, playing = play. Thus reducing the total number of words
- Stop word removal - Stop words are common words with not much value such as 'the', 'a', 'at' etc. Words with length 2 or less were removed.

New results:



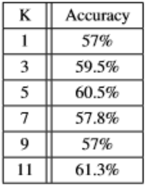
Test #	Train size	Test size	Stemming	Stop words	Accuracy
1	2400	600	Yes	Removed	72%
2	2400	600	No	Included	63.17%
3	1680	420	Yes	Removed	66%
4	1680	420	Yes	Included	66%
5	1680	420	No	Included	57.38%
6	1680	420	No	Removed	56%

Figure: Naïve Bayes results

Results

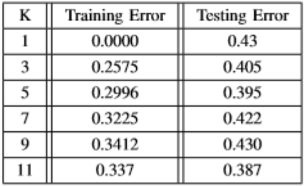
Brought Naïve Bayes accuracy up to 72%, however, stop words removal seems to have less of an effect than stemming.

KNN



K	Accuracy
1	57%
3	59.5%
5	60.5%
7	57.8%
9	57%
11	61.3%

(a) Accuracies per K



K	Training Error	Testing Error
1	0.0000	0.43
3	0.2575	0.405
5	0.2996	0.395
7	0.3225	0.422
9	0.3412	0.430
11	0.337	0.387

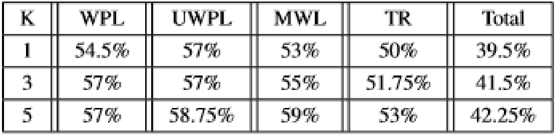
(b) Errors

Results - cont.

KNN results were not so great and seemed to fluctuate with increasing K. This could be due to the overlapping and closeness of data points as seen in the plot.

Feature importance

So how much influence does each feature have?

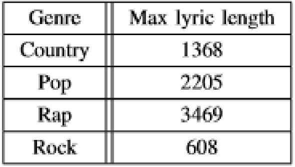


K	WPL	UWPL	MWL	TR	Total
1	54.5%	57%	53%	50%	39.5%
3	57%	57%	55%	51.75%	41.5%
5	57%	58.75%	59%	53%	42.25%

Figure: Table showing accuracies for each removed feature

The removal of total words has the largest effect on the accuracy. The total number of words varies between genres. Rap has quite a high word count where rock has much less.

Feature importance-cont.



Genre	Max lyric length
Country	1368
Pop	2205
Rap	3469
Rock	608

Figure: Table showing songs with highest word counts per genre

Comparisons

- Overall, the Naïve Bayes algorithm performed better than the KNN in both prediction accuracy as well as time performance.
- Bag-of-words was an efficient representation
- The feature extraction proved to provide some useful information on the genre.
- Top accuracies per algorithm : 61.3% for KNN and 72% for Naïve Bayes

Improvements

- Extract more features such as a sentiment score
- Try lemmatization instead of stemming
- Use a built in library of pre-defined stop words for removal

References

[Walsten and Orth] Doran Walsten and Daivik Orth. Song genre classification through quantitative analysis of lyrics. *Unpublished manuscript*.