Genre Classification using Feature Extraction and Deep Learning Techniques

Conference Paper · November 2018

DOI: 10.1109/KSE.2018.8573325

3 authors, including:

Arjun Rajpal
Delhi Technological University
3 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Dushyant Rathore
Delhi Technological University
3 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Genre Classification using Feature Extraction and Deep Learning Techniques

Akshi Kumar

Department of Computer Science and Engineering
Delhi Technological University
Delhi, India
akshikumar@dce.ac.in

Arjun Rajpal, Dushyant Rathore

Computer Science and Engineering
Delhi Technological University
Delhi, India
rajpal.arjun@yahoo.in, dushyant.bgs@gmail.com

***Abstract*— Music genre refers to categorisation of music on the basis of interaction between artists, market forces, and culture. It helps to organize music into collections by indicating similarities between compositions or musicians. Automatic genre classification is non-trivial as it is difficult to distinguish between different genres. Many times the boundaries are not clearly defined and genres are overlapping. In this paper we present a novel approach to classify a list of songs present on Spotify into mainly four genres - Christian, Metal, Country, Rap. Two different kinds of data - lyrics and album artwork are used for the classification process. Two Natural Language Processing techniques namely Bag-Of-Words and Term Frequency-Inverse Document Frequency (TFIDF) are used to process the lyrical data. We apply machine learning algorithms like Random Forest, Support Vector Machine (SVM), Naive Bayes, Linear Support Vector Classifier (Linear SVC) and eXtreme Gradient Boosting (XGBoost) on lyrical data and Deep Convolutional Neural Network (CNN) on the album artwork to predict the genre. On application of machine learning algorithms on lyrical data obtained from Bag-Of-Words a mean precision of 75.96% and a mean f-score of 75.92% is achieved. On application of the same set of algorithms on lyrical data obtained from TFIDF a mean precision of 76.85% and a mean f-score of 77.38% is achieved. In both cases XGBoost outperforms all the other algorithms giving a maximum precision of 79.30% and 80.16% and a maximum f-score of 79.6% and 84.09% for Bag-Of-Words and TFIDF respectively. On application of deep neural network on album artwork, a precision of 82.46% and a f-score of 81.84% is achieved.**

***Keywords* —- Album Artwork; Bag-of-Words; Convolutional Neural Network (CNN); Genre Classification; Lyrics; Term Frequency-Inverse Document Frequency (TFIDF)**

I. INTRODUCTION

Genre classification involves designating objects to a group on the basis of similarities in subject, style, purpose or format. It has been used in music for information management and topic classification. However, the notion of genre is not well-defined for text. There exists some controversy over the constitution of genre among the classification community. Many times, it is possible that a song may belong to more than one genre.

Genre classification by lyrics is a classic example of a Natural Language Processing (NLP) problem [1]. In NLP, the aim is to extract meaning from the text and assign labels to it; here this equates to the genre classification of the lyrical text.

At present, digital music services like Spotify, etc., classify music into sub-genres using song metadata like the digital signatures for some factors such as acoustics, dance ability and emotional tone [2], [3]. The digital music services ignore lyrics as it is difficult to collect large scale lyrical data [4]. However, genre is closely related to lyrics. Genre of the song can be recovered from word frequencies in lyrics.

We try to explore the field of genre classification by applying and comparing the performances of machine learning models like Naive Bayes, Support Vector Machines, XGBoost, etc. on the lyrical data obtained using feature extraction techniques such as Bag-of-Words and TFIDF [5]. In addition to this, we present a relatively new technique of automatic genre classification through application of convolutional neural networks on the album artwork. In recent years, the use of deep learning methods such as Convolutional Neural Networks (CNNs) has led to significant performance improvements and breakthroughs in the field of image classification. While linear and kernel models rely on good hand-selected features these deep learning architectures circumvent this by letting models learn important features themselves.

The rest of the paper is structured as follows. Sections II and III present a literature survey along with the experimental design. Section IV consists of the description of the proposed architecture and machine learning models. Section V gives the research results and visualizations. Finally, section VI presents the conclusion and discusses the future scope of the research.

II. LITERATURE SURVEY

Initial works on automatic genre classification almost exclusively dealt with rhythm and harmony of musical content. In [6], [7], rhythmic content, timbre texture and pitch content were used to evaluate genre.

Later works recognized the importance of lyrics in automatic genre classification. Bag-Of-Words technique established the role of lyrics in genre classification.

In [8], [9], combination of both rhythm based and lyric based classification techniques were used. However only

978-1-5386-6113-0/18/$31.00 ©2018 IEEE

Bag-Of-Words technique was used for textual analysis.

No paper has used more than one method of textual analysis. In this paper, we have used both TFIDF and Bag-Of-Words as NLP techniques and have compared their performances for different machine learning models.

Since the use of album artwork for image classification has not been undertaken previously, we try to explore the field of genre classification using convolutional neural networks on the album art to predict the genres.

We also perform a comparative analysis of the performances of the image data with lyrical data in automatic genre classification.

III. EXPERIMENTAL DESIGN

A. Data Sources and Data Preprocessing

The initial lyrical data is taken from Kaggle [10] and LyricsFreak. The album artwork and genre labels for each song are downloaded using the Spotify API. The final dataset consists of 10,000 songs belonging to the genres - Christian, Metal, Country, and Rap. These four genres are chosen since they are more popular and can be easily differentiated by themes, feature sets, etc. The final dataset is split into 70/30 for training and testing purposes.

Since the Spotify API returns a list of micro-genres for a particular song/artist, one challenge is to determine which broad genre a particular song belongs to. Thus, to determine the target genre, we check if the broad genre (Christian, Metal, Country, and Rap) is a substring of the micro-genres generated for that song by Spotify. After this, the broad genre with the maximum number of instances is selected and used as the target genre for that particular song. Table I describes the attributes present in the dataset.

The lyrical data in the Kaggle dataset is minimally structured and cleaned. To clean it we perform word tokenization and then remove the delimiters, followed by removal of non-alphabetic characters and stop words. Finally, the structured lyrical data is formed by removing the words not present in the English dictionary.

The album artwork needed for genre classification is obtained by using song name and artist name from Kaggle dataset. These details act as inputs to the search function of Spotipy (Spotify's Web API Wrapper for Python) which returns the url address of the album artwork for each song.

B. Lyrical Data

Using the structured/cleaned lyrics available from the preprocessing step, we create two kinds of textual representation structures -

- **Bag-Of-Words** - A vector of length n is used to represent a song. n denotes the count of unique words present in the lyrics of all the cleaned songs. Each feature x(i) in the vector corresponds to the number of times the word i appears in the specific song.
- **TFIDF (Term Frequency-Inverse Document Frequency)** - It is an extension of Bag-Of-Words technique. Each feature x(i) in the vector corresponds to the TFIDF score for the word i in the specified song.

C. Image Data

The album art shown in Fig 1 is resized into a 50*50 gray-scale pixel image. The image is then converted to a feature vector of 2500 pixel values for each song using image matrix transformation techniques. This feature vector is then used as input to our Convolutional Neural Network model.

A Convolutional Neural Network (CNN) [11] is a type of multilayer neural network. It comprises of multiple convolutional layers (often with a sub sampling step) followed by multiple fully connected layers. CNN's architecture makes use of the 2D structure of the input (image, speech, etc.). This is achieved with the help of local connections and tied weights followed by some form of pooling resulting in translation invariant features. CNNs are easier to train and have fewer parameters as compared to fully connected networks having the same number of hidden units.

TABLE I
DATASET ATTRIBUTE DESCRIPTION

Attribute	Description
song	Denotes the name of the song
artist	Name of the artist
lyrics_link	LyricsFreak link containing the lyrics of the song
lyrics	Lyrics of the song
album_uri	Unique album uri on Spotify
artist_uri	Unique artist uri on Spotify
genres	List of micro-genres returned by the Spotify API
album_art	Link containing the album art
target	Target genre for the song obtained by applying the preprocessing step

Fig. 1. Example Metal Song Artwork

IV. PROPOSED ARCHITECTURE

A. Model One : Lyrical Data

*1) **Bag of Words**:* It is an algorithm that counts the number of times a word appears in a document. These word counts help to compare documents and gauge their similarities for applications like topic modeling, document classification and search. It lists the words along with their

word counts per document. E.g.: If there are two sentences –

1) Sam likes to read comics. Maria likes comics too.
2) Sam also likes to read thriller novels.

We create a corpus of all the unique/distinct words present in both the sentences as follows: ['Sam', 'likes', 'also', 'to', 'read', 'Maria', 'comics', 'thriller', 'novels', 'too']. Finally, they represent each sentence as feature vectors of length n (size of the corpus) where feature x(i) in the vector represents the number of times the word i (of the corpus) appears in the specific sentence.

$$BOW1 = [1; 2; 0; 1; 1; 1; 2; 0; 0; 1]$$

$$BOW2 = [1; 1; 1; 1; 1; 0; 0; 1; 1; 0]$$

*2) **TFIDF**:* TFIDF stands for Term Frequency-Inverse Document Frequency. TFIDF weight is a statistical measure used to evaluate the importance of a word to a document in a corpus. It is often used in text mining and information retrieval. The importance of a word increases in direct proportion to the number of times it appears in the document. However, it is neutralized by the frequency of that word in the corpus. It is defined as:

$$tfidf(t, d) = tf(t, d) * idf(t) \quad (1)$$

where: TFIDF(t,d) is TFIDF score of term t in document d, TF(t,d) is number of times the term t appears in document d, and IDF(t) is IDF score of term t.
IDF(t) is defined as:

$$IDF(t) = log((1 + n_d)/(1 + DF(d, t))) + 1 \quad (2)$$

where: n_d is the number of documents, and DF(d,t) is number of documents which have the term t.

We apply L2-Normalization because in the TF (Term Frequency), they represent each term as the number of times it appeared in the document. However, a major problem with this representation is that it creates a bias towards long documents, as a given term has more chances to appear in the longer document, thus making them appear more important than they actually are.

The modified formula for $TFIDF(t_i, d)$ with L2-Normalization is defined as:

$$TFIDF(t_i, d)' = \frac{TFIDF(t_i, d)}{\sqrt[2]{\sum_{i=1}^{m}(TFIDF(t_i, d))^2}} \quad (3)$$

where: m is the no. of unique words in the corpus and TFIDF(t_i ,d) represents the TFIDF score for i^{th} word in the corpus for document d.

After preprocessing the lyrical data using Bag-Of-Words and TFIDF techniques, we apply the following machine learning models with their default parameters —

- **Naive Bayes** : This technique uses Bayes Theorem and probability theory to categorize the sample. The Bayes Theorem is used to get the probabilities and the algorithm returns the category with the highest probability for that sample. The probability of the feature is calculated using the prior knowledge of the related features. The posterior probability P(t|x) is calculated using the following formula:

$$P(t|x) = (P(t) * P(x|t))/(P(x)) \quad (4)$$

where: $P(t|x)$ is the posterior probability of target class, given predictor (x), P(t) is the prior probability of target class, $P(x|t)$ is the likelihood which is the probability of predictor given target class, and P(x) is the prior probability of predictor.
- **Linear SVC** : This is an implementation of Support Vector Classification where the kernel is linear. It works better for a large sample. It gives more choices of penalties and loss functions.
- **Support Vector Machine (SVM)** : These are supervised learning models and their associated learning algorithms. They help in analyzing the data used for classification and regression analysis. In SVM model, the examples are represented as points in space such that different categories are separated by wide gaps. The new examples are mapped in this space. Depending on the side of the gap in which they fall, they are classified into a particular category.
- **Random Forest** : This method is used for regression and classification. It is an ensemble learning model. In this multitude decision trees are constructed during training time. The mode of the classes is outputted and the mean prediction of individual trees is calculated. The use of random decision forests corrects the habit of decision trees to over-fit to the training set.
- **XG Boost (eXtreme Gradient Boosting)** : This refers to Gradient Boosting decision tree which enhances both speed and performance. In this approach new models are created. These models predict the errors of the previous models. These are then added together to arrive at the final prediction. While adding new models XG Boost makes use of gradient descent algorithm to minimize loss.

B. Model Two : Image Data

As described earlier each album art is transformed to a 50x50 gray-scale image. This image is finally converted to a feature vector using image matrix transformation techniques available in Numpy library in python. The converted album arts of the training set are passed into a Convolutional Neural Network (CNN) for classification.

Our convolutional neural network model as shown in Fig 2 consists of 4 convolutional layers having "ReLU" activation function and MaxPool layers. The number of filters for these convolutional layers are 32, 64, 32 and 128 respectively. In addition to the convolutional layers, we make use of a fully connected layer having a "ReLU" activation function and 1024 output units. Finally the output units

from this fully connected layer are fed into another fully connected layer having 4 output units denoting the target genre classes. The parameter values of our Convolutional Neural Network model are shown in Table II.

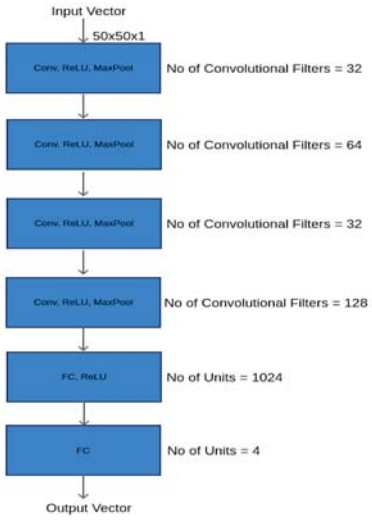

Fig. 2. CNN Architecture

TABLE II
PARAMETER VALUES FOR THE CONVOLUTIONAL NEURAL NETWORK MODEL

Parameter	Values
Epochs	15
Activation function for input and hidden layers	ReLU
Activation function for output layer	Softmax
Loss	categorical_crossentropy
Optimizer	Adam Optimizer (learning rate=0.001, beta1=0.9, beta2=0.999, epsilon=1e-08, use_locking=False)
Dropout	0.8

V. RESEARCH RESULTS

A. Model One : Lyrical Data

On application of the above-mentioned machine learning algorithms, we obtain the precision and f-score of each of Bag-of-words and TFIDF model as shown in Table III and Table IV. As one can make out from Table III and Fig 3 and Table IV and Fig 4, the TFIDF textual analysis model outperforms the Bag-Of-Words model in 4 of the 5 machine learning models in terms of both precision and f-score. On further observation, we find that XGBoost algorithm gives the maximum precision of 79.3% and maximum f-score of 79.6% in Bag-of-words and maximum precision of 80.16% and maximum f-score of 84.09% in TFIDF analysis models. This supports the established domination of the XGBoost algorithm in the field of applied machine learning for structured data.

TABLE III
PRECISION SCORE FOR GENRE CLASSIFICATION

Classifiers	Bag-of-words %	TFIDF %
Naive Bayes	75.0	68.5
Linear_SVC	70.7	79.2
SVM	77.1	77.4
Random Forest	77.7	79.0
XGBoost	79.3	80.16

TABLE IV
F-SCORE FOR GENRE CLASSIFICATION

Classifiers	Bag-of-words %	TFIDF %
Naive Bayes	75.8	66.7
Linear_SVC	70.1	78.2
SVM	76.3	78.4
Random Forest	77.8	79.5
XGBoost	79.6	84.09

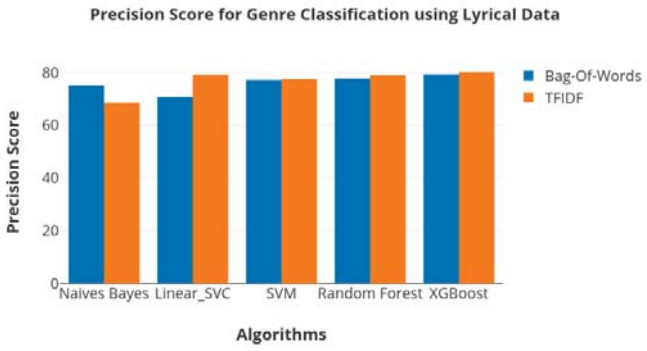

Fig. 3. Precision Score for different models

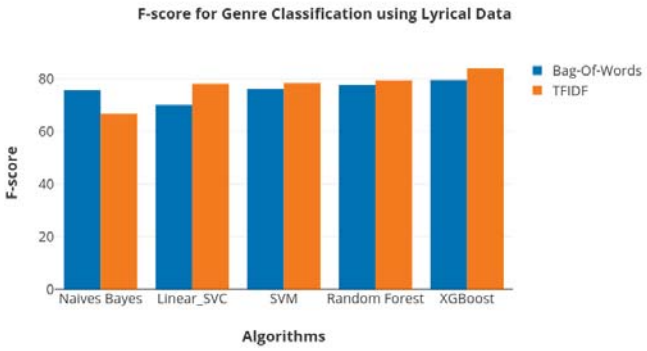

Fig. 4. F-score for different models

B. Model Two : Image Data

On application of our Convolutional Neural Network model, we obtain a precision of 82.46% and f-score of 81.84%. Variation in accuracy, validation accuracy and log loss with change in epochs is shown in Fig 5 and Fig 6. It is evident from the figures that the model accuracy increases and the log loss decreases with the subsequent epochs.

The normalized and un-normalized confusion matrix for the predictions made by our model are shown in Fig 7 & Fig 8 respectively.

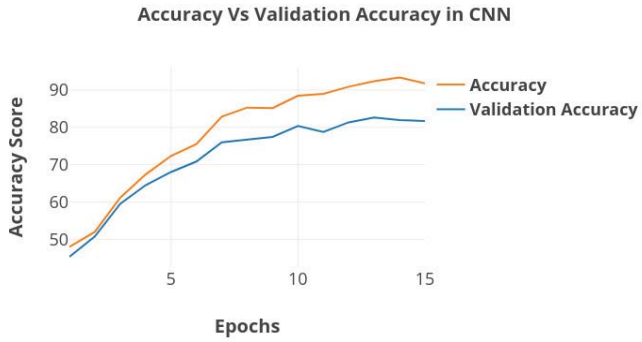

Fig. 5. Accuracy Vs Validation Accuracy in CNN

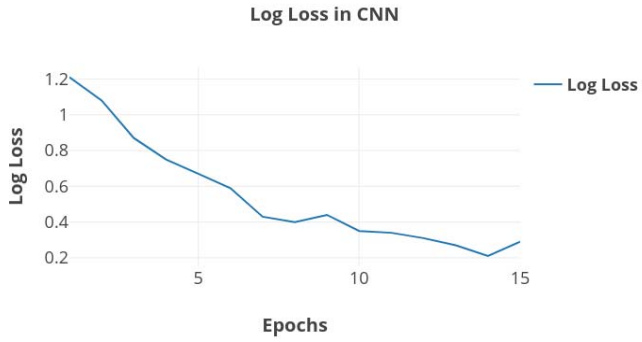

Fig. 6. Log Loss in CNN

VI. CONCLUSIONS

While performing automatic genre classification on songs, it is found that both lyrics and album artwork have a deep influence on determining the genre of a particular song. It is also discovered that since TFIDF, unlike Bag-Of-Words, not only takes into account the frequency of a particular word in a particular song but also considers its frequency in the entire corpus and assigns terms weights to each word, so it outperforms the Bag-Of-Words technique in case of lyrical data. Also, it is observed that there is a huge possibility for more exploration of album artwork in regards to genre and other information about the song.

For future work, we plan to explore more sophisticated

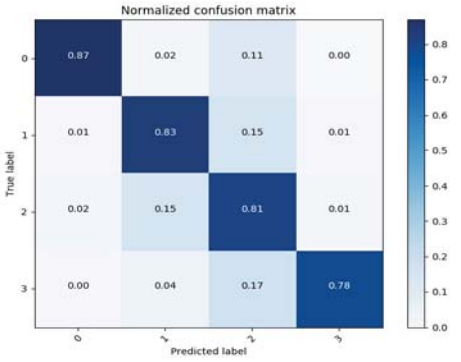

Fig. 7. Normalized Confusion Matrix

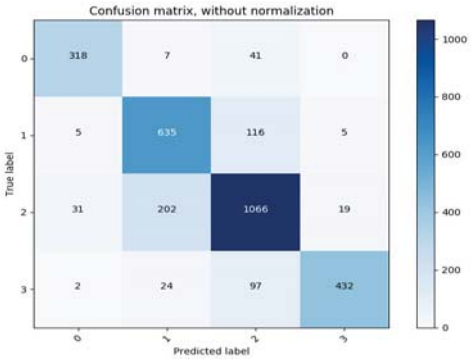

Fig. 8. Un-normalized Confusion Matrix

natural language processing techniques like word2vec, doc2vec and semantic analysis [12] for rigorous cleaning of the lyrical data. Along with it, classifying the songs in an even wider range of genres by using other features like beats per minute, tone patterns, etc can also have a significant effect on the domain. Finally, applying a suitable optimization technique like Particle Swarm Optimization with an objective of maximizing both precision and f-score in case of both lyrical data and album artwork can also lead to significant improvements in the results.

REFERENCES

[1] Mahedero et al., "Natural language processing of lyrics," in Proceedings of the 13th Annual ACM International conference on Multimedia, MULTIMEDIA'05, November 06 - 11, 2005, Hilton, Singapore, doi:10.1145/1101149.1101255.

[2] T. Dammann and K. Haugh, "Genre Classification of Spotify Songs using Lyrics, Audio Previews, and Album Artwork".

[3] Teh Chao Ying, S. Doraisamy and Lili Nurliyana Abdullah, "Genre and mood classification using lyric features," in Proceedings of 2012 International Conference on Information Retrieval & Knowledge Management, March 13 - 15, 2012, Kuala Lumpur, Malaysia, pp. 260-263, doi:10.1109/InfRKM.2012.6204985.

[4] D. Liang, H. Gu, and B. O'Connor., "Music Genre Classification with the Million Song Dataset".

[5] X. Hu, J.S. Downie, & A.F. Ehmann, "Lyric Text Mining in Music Mood Classification.," in Proceedings of 10th International Conference on Music Information Retrieval, ISMIR 2009, October 26 - 30, 2009, Kobe, Japan, pp. 411-416.

[6] G. Tzanetakis and P. Cook, ”Musical genre classification of audio signals,” IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293-302, July 2002.
[7] N. Scaringella, G. Zoia and D. Mlynek, ”Automatic genre classification of music content: a survey,” IEEE Signal Processing Magazine, vol. 23, no. 2, pp. 133-141, March 2006.
[8] R. Neumayer and A. Rauber, ”Integration of Text and Audio Features for Genre Classification in Music Information Retrieval.,” in Proceedings of 29th European Conference on IR Research, ECIR 2007, April 2-5, 2007, Rome, Italy, pp. 724-727, doi:10.1007/978-3-540-71496-5_78.
[9] R. Mayer and A. Rauber, ”Building ensembles of audio and lyrics features to improve musical genre classification,” in Proceedings of 2010 International Conference on Distributed Frameworks for Multimedia Applications, Yogyakarta, 2010, pp.1-6.
[10] S. Kuznetsov, ”55000+ Song Lyrics, Kaggle,” 2016, URL: https://www.kaggle.com/mousehead/songlyrics.
[11] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” arXiv:1511.08458 [cs.NE], November. 2015
[12] B. Logan, A. Kositsky and P. Moreno, ”Semantic analysis of song lyrics,” 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763), Taipei, 2004, pp. 827-830 Vol.2. doi: 10.1109/ICME.2004.1394328