August 2021

Analysis of Music Genre Clustering Algorithms

Samuel Walter Stern
University of Wisconsin-Milwaukee

ANALYSIS OF MUSIC GENRE CLUSTERING ALGORITHMS

by

Samuel Stern

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Computer Science

at

The University of Wisconsin-Milwaukee

August 2021

ABSTRACT

ANALYSIS OF MUSIC GENRE CLUSTERING ALGORITHMS

by

Samuel Stern

The University of Wisconsin-Milwaukee, 2021
Under the Supervision of Professor Susan McRoy

Classification and clustering of music genres has become an increasingly prevalent focus in recent years, prompting a push for research into relevant algorithms. The most successful algorithms have typically applied the Naive Bayes or k-Nearest Neighbors algorithms, or used Neural Networks to perform classification. This thesis seeks to investigate the use of unsupervised clustering algorithms such as K-Means or Hierarchical clustering, and establish their usefulness in comparison to or conjunction with established methods.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank Professor Susan McRoy, whose insightful and timely direction proved critical in helping me navigate every step of the research and academic writing process.

I would also like to voice my gratitude for Professors John Boyland and Henry Trimbach at the University of Wisconsin-Milwaukee and Professor Paul Wilson at the University of Wisconsin-Madison for helping me find my bearings in academic and research pursuits.

Finally, I want to offer my deepest thanks to my father, Avraham Stern, without whose support I could not have persevered through the trials of college and of recent life, and to my late mother, Elizabeth Stern, who I have striven to make proud in my journey through higher education.

INTRODUCTION

The classification and clustering of audio and music represents a significant focus for many researchers, many motivated by the ubiquity of such clustering in content-recommendation algorithms, as well as a push to improve general audio recognition software. Several content-hosting sites have drawn scrutiny for the perceived quality (or lack thereof) of their recommendation algorithms, and the spread of voice-controlled devices (including Amazon's Echo product line, hands-free car media, and others) clearly demonstrate the recent push to improve the quality of automated audio analysis. There are also active investigations in the use of stylistic analysis to settle disputes of authorship or IP. In these cases, algorithms (most frequently under the Machine Learning umbrella) are applied to compare the similarities between contested works and those by all potential true authors (Brinkman, A & Shanahan, Daniel & Sapp, Craig, 2016). Additionally, many popular methods rely on large pre-labeled databases for training, which may not always be reliable or available.

In order to further these efforts, this thesis aims to investigate deeper into the utility of two approaches – the K-Means and Hierarchical clustering algorithms – for the purposes of music classification and insight. Both of these algorithms are 'unsupervised' - they operate without an initial set of labeled data, which makes them more versatile than the popular supervised methods and means they may be used to generate initial datasets for rapid automated training of supervised methods. The two algorithms will be applied to a collection of songs in order to evaluate how accurately they can gather songs of like

genre into the same cluster. These algorithms will then be compared against the classification results for a competing method, the K-Nearest Neighbors algorithm. In this way, this thesis aims to determine their effectiveness for the task of genre clustering, the means by which this effectiveness can be improved, as well as the properties that may make these methods preferable to existing alternatives.

BACKGROUND

Dataset and Features

Methods and Features for Describing and Analyzing Music

There are a wide array of features that are used for the analysis of a musical track, from low- to high-level. The lowest-level features center around direct analysis of the audio spectrogram that is used to describe the amplitudes and frequencies of sound emitted in the playing of music. These features have the advantage of always being available as long as an audio recording is also available. However, these features are also vulnerable to the presence of background noise, and also produce so much data that narrowing a spectrogram's features down to only those that are useful can prove challenging. By contrast, high-level features aim to describe music in terms that are typically used by or are applicable to humans. These features may include a wide array of information such as sequences of notes, the tempo of a song's beat, and sentimental analysis of a song's lyrics or title. As these features require information presented in terms of human perception, however, they must almost always be provided directly by a

human and are challenging to derive from an audio input directly. High-level features are therefore harder to obtain than low-level features, and datasets that include them are often smaller than datasets containing exclusively low-level data.

Dataset: The Free Music Archive

The Free Music Archive (FMA) is an assembly of over 100,000 clips of 30 seconds from music tracks as well as pre-extracted data from them, including a hierarchy of 161 total genres derived from 16 top-level categories (Defferard, M. Benzi, K. Vandergheynst, P. Bresson, X, 2017). The archive contains both high-level audio features of approximately 13,000 of its tracks through the Spotify API, and a collection of low-level features from all clips collected directly from each clip's audio spectrogram in the form of Mel-Frequency Cepstral Coefficients.

Mel-Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients (MFCCs) are a means of describing the low-level waveform data of a sound in a configuration that is intuitive and useful for analysis (Logan, B, 2000). The Mel scale is a frequency scale whose values are adjusted to better match the pitch of a sound experienced by a human, allowing automated analysis to more easily focus on the aspects of an audio track that are of interest to human listeners. The pitch experienced by a human is of a logarithmic relationship to the actual frequency of the sound wave, and further coefficients to describe the relationship were found experimentally. Mels (the unit used by the Mel scale) are related to Hertz by equation (1).

$$M(f) = 1125 \ln\left(1 + \frac{f}{700}\right) \quad (1)$$

Once an audio signal has been converted to the Mel-Scale, the signal is then partitioned into subsections called ‘frames’ consisting of about 20-25ms each. A ‘cepstrum’ is then extracted from each frame by performing a Fourier transform of the frame’s frequency spectrum, taking the log of this transform, and then applying an inverse Fourier transform to the result. This process is displayed in equation (2).

$$C(s(t)) = F^{-1}\left[\log\left(F\left[s(t)\right]\right)\right] \quad (2)$$

The conversion from spectrum to cepstrum isolates information about the relative volume of individual frequency levels within an audio signal. This allows researchers to isolate the particular pitch or pitches that are dominant during any particular frame.

The coefficients that define the frame’s cepstrum on the Mel scale (the Mel-Frequency Cepstral Coefficients) are therefore representative of the pitches and tones of that frame. Since all values have been converted to the Mel scale, each set of coefficients can be treated as a numerical representation of the notes played in a song, with the differences in values between MFCCs approximating the experienced difference between notes by a human listener. There are theoretically infinite MFCCs that can be used to describe any given signal (given that they are derived from a Fourier-transform-like operation), but it has been determined that the first 12-13 are almost always the best performing and most useful, and that the contributions of further values are negligible (Seyed Reza Shahamiri, Siti Salwah Binti Salim, 2014).

As there are hundreds of frames present in a 30-second audio clip, the FMA dataset provides its MFCC information in the form of statistical data about each MFCC

across its clip's frames, rather than providing each frame's MFCCs directly. The means, standard deviations, skews, and kurtosis values are provided for each of the first 20 MFCCs across each clip, though only the first 13 of each such value will be investigated for this thesis.

High-level Audio Features

In addition to the low-level MFCCs, which describe sounds directly, there are a number of high-level features available through the Spotify API (Long, M., Hu, L., & Jin, F, 2021). Of interest to this thesis are the following eight features, as described on the Spotify API web page:

- **Acousticness:** "A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic. "
- **Danceability:** "Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable."
- **Energy: "**Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy."
- **Instrumentalness:** "Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content."

- **Liveness:** “Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.“
- **Speechiness:** “Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.“
- **Tempo:** “The overall estimated tempo of a track in beats per minute (BPM).”
- **Valence:** A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Of note is that some datasets (such as the FMA dataset, which will be used in this thesis) have also obtained these features from the Echo Nest API, which has since been merged into the Spotify API after Spotify’s acquisition of Echo Nest.

Existing Approaches

K-Nearest Neighbors

K-Nearest neighbor (k-NN) classification requires an existing body of samples whose class is already known and a user-selected value k. Then, when a new sample is input, the k-NN classifier categorizes it into whatever class appears the most among the k nearest neighbors to the new sample. This newly-categorized sample is then added to the body of known samples, and the process is repeated until all samples have been

classified. In a 2004 study by Li and Sleep, a 1-Nearest Neighbor algorithm consistently predicted music genres with over 85% accuracy.

The supervised k-NN algorithm has a natural advantage to its accuracy over unsupervised clusterers by virtue of possessing training data in advance. As a result, additional experiments are required in order to correct for this advantage and obtain a clear picture of how the k-NN algorithm compares with the K-Means and Hierarchical models.

Naive Bayes

The Naive Bayes approach hinges on Bayes' Theorem, displayed in equation (3), is the basis of a successful approach for music genre classification.

$$P(A \vee B) = \frac{P(B \vee A) * P(A)}{P(B)} \quad (3)$$

One can apply this theorem to determine the probability of a class given the truth value of another class. When provided with a large corpus of data and appropriate preprocessing, the Naive Bayes approach has predicted music genres with over 90% accuracy (Kofod, C., & Ortiz-Arroyo, D, 2008).

Deep Learning Approaches

Deep Learning is very accurate, but tends to be less efficient and is critically poor at showing more data than its specific output. Techniques of this type often center around the use of Convolutional Neural Networks. These networks implement 'convolutional' layers, which isolate subsections of an input in order to identify high-level features, such

as temporally related data (Yu Zheng, Quiuyu Chen, Jianping Fan, Xinbo Gao; 2020). These convolutional layers then feed forward into further neural network layers, which are then able to perform their calculations with clearer virtual pictures of the high-level features.

While these methods are often highly effective (achieving 91% accuracy in a 2017 study by Senac et. al) they can often be costly to train and require a large training dataset before they can begin to effectively analyze individual samples. In addition, the training of neural network nodes is a relative ‘black box’, which provides little information about how it internally organizes its classifications or the data which led to them.

Unsupervised Clustering Algorithms

The K-Means Algorithm

The K-Means algorithm begins with a distribution of samples and K initial cluster centers, which are either manually chosen by a user or are automatically selected by other algorithms. Each sample is then systematically sorted into the cluster whose centroid is nearest to the sample, moving the centroid in the process. This technique is considered a ‘top-down’ or ‘divisive’ approach, because it begins with all samples essentially in one ‘undetermined’ cluster, and then divides them into its K clusters.

This algorithm is relatively efficient, but requires foreknowledge of the number of target clusters to achieve, and often does little to illustrate the relationships between its clusters (such as sub-genre relationships or music that can be considered close to several genres). In addition, the sorting order of samples into clusters must be carefully (and

often inefficiently) optimized as outliers can cause a centroid to move to an inappropriate location, damaging the algorithm's ability to accurately group other samples with its cluster.

Hierarchical Clustering

As an alternative to other methods, the hierarchical approach instead initializes each sample as being its own 'cluster', and then iteratively joins the nearest two clusters until all clusters are either joined into a single super-cluster, or are a requisite distance away from each other. Because of this process of gradually joining clusters (as opposed to splitting them up), this technique is considered 'bottom-up' or 'agglomerative' clustering. Unlike K-Means clustering, Hierarchical clustering has no requirement of foreknowledge of the cluster count. In addition, as the sub-clusters to any given cluster can be visualized on a dendrogram, this method allows it to evaluate sub-genre relationships and evaluate the closeness of any two genres (and, naturally, their constituent songs).

Feature Selection

Motivation

The FMA dataset provides four different statistical features (mean, standard deviation, skew, and kurtosis) for each of the first 20 MFCCs extracted from each audio clip, in addition to the eight high-level Spotify API features. Of these, many features are irrelevant to the task of genre-based clustering, and may worsen results by adding 'noise' to the clustering process. In addition, clustering algorithms grow inefficient when

presented with excessive variables. In order to avoid these problems, the process of feature selection is applied in order to trim out features that contribute negatively or negligibly to the accuracy of the experimental algorithms. In this experiment, a genetic algorithm was employed for feature selection due to its ability to accommodate large numbers of potential features and to evaluate them together which is important for capturing possible dependencies among features.

Genetic Algorithms

In this strategy, generations of a simulated population are produced, in which each available feature may be toggled on or off in each entity in the population. The population is then assessed for a level of ‘fitness’, and the fittest entities undergo a simulated ‘reproduction’ in which their selected features are combined with other fit entities and passed on to the next generation. This proceeds until a maximum of generations is reached or until the progressing of a generation fails to increase the fitness above a certain relative threshold. For the purposes of this experiment, the fitness function was the score of a k-NN classifier run using selected subsets of the feature options, in order to gain a direct picture of the quality of the features for the methods under examination.

Metrics

The Fowlkes-Mallows Index

The Fowlkes-Mallows Index (FMI) is the geometric mean of the precision and recall of a set of classified points (Halkidi, Maria; Batistakis, Yannis; Vazirgiannis,

Michalis, 1 January 2001). Unlike many similar units, however, the FMI measures precision and recall in a pairwise fashion; that is, it defines ‘true positive’ as an instance where a pair of samples that belong in the same class are placed in this class, ‘false positive’ as an instance where a pair of samples are predicted to share a class but do not share it in actuality, and ‘false negative’ as a case where two samples are not classified to the same class, but should have been. For a count of true positives P_t, false positives P_f, and false negatives N_f, the FMI is defined by equation 4:

$$FMI = \frac{P_t}{\sqrt{(P_t + P_f) * (P_t + N_f)}} \tag{4}$$

In this way, it measures success in a manner that is agnostic of the actual class labels, and therefore may be used to score unsupervised clusterers, which produce their own class labels based on the connections between samples.

METHODOLOGY

For this experiment, the features deemed potentially relevant were the eight high-level audio features from the Spotify API as well as the means and standard deviations of first 13 MFCCs. These features were then used as traits in a genetic algorithm set to score its populations’ fitness using k-NN classification for scoring in order to determine the optimal features from those available. This was repeated for varied five different maximum feature counts (30, 20, 15, 10, and 5) in order to best establish the optimal feature set. Each iteration of this genetic algorithm was applied with a population equal to ten times the maximum feature count and 50 generations of evolution.

A corpus of the 5,700 tracks from the FMA dataset were then assembled from the 6 most populous genres for which the dataset had complete information. This body of data was then clustered using both the K-Means and Hierarchical approach, as well classified using a k-NN approach with five different K-values. These K-values were set at 1, 5, 10, 50, and 100, in order to capture the accuracy of the approach at varied orders of magnitude. As K-Means and Hierarchical algorithms are both unsupervised clusterers, they do not directly label the classes of their final cluster outputs, so the FMI is used as a metric to measure the accuracy of all experimental algorithms.

Once the ideal number of features has been identified, the k-NN algorithm was then run at decreasing proportions of training to testing data, with the goal of determining its behavior as its training data decreases.

RESULTS & DISCUSSION

Feature Selection

Feature Selection Results

Given each described maximum for feature count, the genetic algorithm selected features from the available high-level features, as well as the statistical summaries of specific coefficients from the MFCCs. The selected coefficients and audio features, as well as the cross-validation scores for the genetic algorithm upon making such selections, are listed in Table 1.

Maximum Features	MFCC Means	MFCC Standard Deviations	MFCC Skews	MFCC Kurtosis	High-Level Audio Features	CV Score
30	2, 3, 4, 5, 6, 7, 8	1, 2, 6, 8, 9	3, 4, 6, 7, 10	3, 5, 7, 9	Speechiness, Danceability, Instrumentalness, Energy, Acousticness, Valence	0.773
20	2, 3, 4, 5, 6, 7, 8	1, 6, 8	6, 7, 8	3, 7	Speechiness, Danceability, Instrumentalness, Energy, Valence	0.773
15	2, 3, 4, 5, 6	6	7, 10	3	Speechiness, Danceability, Instrumentalness, Energy, Valence	0.759
10	2, 3, 4	1, 6, 8	-	-	Speechiness, Danceability, Instrumentalness, Energy	0.751
5	2, 3	8	-	-	Speechiness, Instrumentalness	0.730

Table 1: Features selected by the genetic algorithm at various settings for maximum feature count.

Feature Selection Discussion

Of note is that, when the maximum feature count was set to 30, only 27 features were selected. As a result, no higher maximum feature counts were used. Of further note is that, while most decreases in maximum feature count resulted in the removal of features previously selected, the changes between 15, 10, and 5 features caused some MFCC values' standard deviations to return, and while the standard deviation of MFCC 6 was selected at 15 features, 8 was chosen instead at 5. This may be due to a difference in the evolutionary path chosen by the genetic algorithm (indicating that these values are of approximately equal value) or that they may be correlated with other values whose selections had also been changed.

Classifier Testing

Classifier Testing Results

The five k-NN algorithms and two clustering algorithms were then used to classify and cluster the genres of the tracks using the selected features. The resulting Fowlkes-Mallows score for each algorithm at each feature count is displayed in Table 2.

Feature Count	1-NN	5-NN	10-NN	50-NN	100-NN	Hierarchical	K-Means
27	0.601	0.657	0.660	0.659	0.658	0.494	0.450
20	0.590	0.649	0.662	**0.670**	0.659	0.471	0.449
15	0.578	0.636	0.648	0.644	0.644	0.502	0.461
10	0.565	0.633	0.645	0.634	0.634	0.457	0.465
5	0.516	0.593	0.622	0.618	0.619	0.435	0.510

Table 2: FMI of classifiers and clusterers for varied feature counts.

Classifier Testing Discussion

The k-NN algorithm performed best at high k-values in this problem, though the difference between k=10, k=50, and k=100 was consistently small enough to be attributable to random noise. There is no discernible pattern clearly connecting the FMI of the clustering algorithms to the feature count. As all of the features were optimized for a k-NN classifier, this may indicate that the features which are optimal for a k-NN classifier are not also optimal for unsupervised clustering algorithms.

The two clustering algorithms, while consistently outperformed by the k-NN algorithm, showed encouraging results; for the lowest feature count, the K-Means model was nearly equal in quality to the 1-Nearest Neighbor classifier. It may therefore be possible that, given more optimally selected features (or simply a small number of

features), an unsupervised clustering algorithm may be developed that can consistently produce comparable results to those of a supervised method.

Impacts of Dataset Size

Training Set Size Results

In order to more fully explore the comparability of quality between the unsupervised clusterers and the supervised k-NN algorithm, learning curves were generated for the k-NN algorithms. The results of this test are compiled in Table 3 and illustrated in Chart 1.

Training Set Size	1-NN	5-NN	10-NN	50-NN	100-NN
4000	0.5901	0.649	0.662	0.6704	0.659
2850	0.5799	0.637	0.648	0.663	0.661
1700	0.563	0.617	0.634	0.647	0.643
570	0.533	0.591	0.618	0.632	0.623
285	0.512	0.586	0.604	0.607	0.587
170	0.502	0.563	0.596	0.595	0.5703
114	0.469	0.553	0.582	0.584	0.621
57	0.435	0.515	0.534	0.526	-
28	0.458	0.505	0.509	-	-

Table 3: FMI of k-NN classification on 5700-sample dataset at varied training set size.

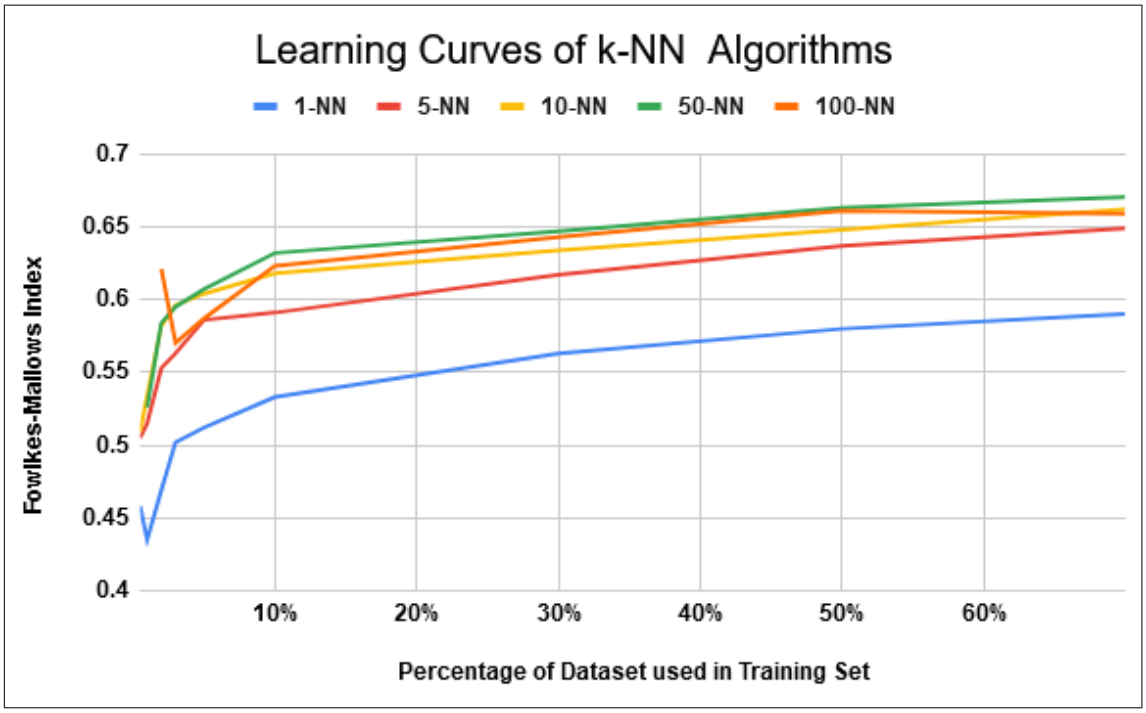

Chart 1: Learning Curves of k-NN algorithms for varied values of K.

Training Set Size Discussion

The results show that the k-NN algorithm experiences diminishing returns on accuracy as the training set size increases, and conversely is prone to becoming exponentially more inaccurate as training set size decreases. Notably, when the training set is of less than 100 elements, all results from k-NN models begin to approach those of the K-Means and Hierarchical models. This follows logically, as the clustering algorithms behave very similarly to the k-NN algorithm but merely operate without a predefined body of labeled data, then in the absence of labeled data, the k-NN algorithm should converge toward the same results as the unsupervised clusterers.

Of note, however, are the increases in accuracy of the 1-NN and 100-NN models with their smallest training set sizes. This (as one would expect of a small set size) would

indicate that some manner of random variance or volatility may be influencing the results of those tests.

CONCLUSION

In this experiment, the k-Nearest Neighbors, K-Means, and Agglomerative clustering models were used to cluster a dataset of music tracks by genre under varied sets of features and conditions. In this way, data was accrued that allowed the three methods to be compared, so that the viability of the rarely-used unsupervised clustering algorithms could be determined relative to the oft-used k-NN classifier. The feature selection process seemed to favor the high-level audio features, indicating that there is merit in using them, even if the means of their derivation is not publicly accessible. In addition, the means of the MFCC values were clearly more frequently selected than any other statistical information, and the standard deviations were most consistently chosen among the remaining features when prohibitively small feature limits were placed.

The best result was a 67% FMI, achieved using a 50-Nearest Neighbor classifier with 20 features from this dataset, with 70% of the dataset used for training. As a 2013 study conducted by Velarde et. al achieved a k-NN accuracy of 88% on a corpus of 26 different genres, it is reasonable to conclude that their methodology and selected features represent a significantly higher baseline for the quality of this method. As such, there may be cause to replicate the features and steps of their experiment for the evaluation of the unsupervised clusters as well, in order to attempt to establish a higher quality baseline for them.

The tendency of k-NN to approach the accuracy of the unsupervised methods when faced with small training sets may warrant further investigation, as it would imply there may be merit to producing variants of the hierarchical and K-means algorithms that can be ‘seeded’ with initial clusters. If these algorithms follow the tendency of the k-NN model, then that would indicate that even a relatively small amount of initial information (in this case, as little as 3% of the dataset’s total size) may drastically improve the results.

REFERENCES

Brinkman, A & Shanahan, Daniel & Sapp, Craig. (2016). Musical Stylometry, Machine Learning, and Attribution Studies: A Semi-Supervised Approach to the Works of Josquin. 91-97.

Débora C. Corrêa, Francisco Ap. Rodrigues, A survey on symbolic data-based music genre classification, Expert Systems with Applications, Volume 60, 2016, Pages 190-210, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2016.04.008.(https://www.sciencedirect.com/science/article/pii/S095741741630166X)

Defferard, M. Benzi, K. Vandergheynst, P. Bresson, X. (2017) FMA: A Dataset for Music Analysis (https://arxiv.org/abs/1612.01840)

Halkidi, Maria; Batistakis, Yannis; Vazirgiannis, Michalis (1 January 2001). "On Clustering Validation Techniques". Journal of Intelligent Information Systems. ***17*** *(2/3): 107–145. doi:10.1023/A:1012801612483.*

Kofod, C., & Ortiz-Arroyo, D. (2008, December). Exploring the design space of symbolic music genre classification using data mining techniques. In *2008 International Conference on Computational Intelligence for Modelling Control & Automation* (pp. 43-48). IEEE.

Li, M., & Sleep, R. (2004). Melody classification using a similarity metric based on kolmogorov complexity. *Sound and Music Computing*, *2012*.

Logan, B. Mel Frequency Cepstral Coefficients for music modeling. First International Symposium on Music Information Retrieval. (http://ciir.cs.umass.edu/music2000)

Long, M., Hu, L., & Jin, F. (2021, March). Analysis of Main Characteristics of Music Genre Based on PCA Algorithm. In *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)* (pp. 101-105). IEEE.

Manuel Calzolari. (2021, April 3). manuel-calzolari/sklearn-genetic: sklearn-genetic 0.4.1 (Version 0.4.1). Zenodo. http://doi.org/10.5281/zenodo.4661248

Nadeau, D. Sekine, S. (2007, August) A Survey of Named Entity Recognition and Classification. (https://nlp.cs.nyu.edu/sekine/papers/li07.pdf)

Panda, R., Redinho, H., Gonçalves, C., Malheiro, R., & Paiva, R. P. (2021, July). How Does the Spotify API Compare to the Music Emotion Recognition State-of-the-Art?. In *Proceedings of the 18th Sound and Music Computing Conference (SMC 2021)* (pp. 238-245). Axea sas/SMC Network.

Senac, C. Pellegrini, T. Mouret, F. Pinquier, J. 2017. Music Feature Maps with Convolutional Neural Networks for Music Genre Classification. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing* (*CBMI '17*). Association for Computing Machinery, New York, NY, USA, Article 19, 1–5. DOI:https://doi.org/10.1145/3095713.3095733

Seyed Reza Shahamiri, Siti Salwah Binti Salim, Artificial neural networks as speech recognisers for dysarthric speech: Identifying the best-performing set of MFCC parameters and studying a speaker-independent approach, Advanced Engineering Informatics, Volume 28, Issue 1, 2014, Pages 102-110, ISSN 1474-0346, https://doi.org/10.1016/j.aei.2014.01.001. (https://www.sciencedirect.com/science/article/pii/S1474034614000020)

Velarde, G., Weyde, T., & Meredith, D. (2013). An approach to melodic segmentation and classification based on filtering with the Haar-wavelet. *Journal of New Music Research*, *42*(4), 325-345.

Yu Zheng, Qiuyu Chen, Jianping Fan, Xinbo Gao, Hierarchical convolutional neural network via hierarchical cluster validity based visual tree learning, Neurocomputing, Volume 409, 2020, Pages 408-419, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2020.05.095. (https://www.sciencedirect.com/science/article/pii/S092523122030970X)