Investigating music genres: An unsupervised music clustering approach

Rafael A. Mayer
rmayer@ifi.unicamp.br

Instituto de Física Gleb Wataghin (IFGW) Universidade Estadual de Campinas (Unicamp)
Campinas, SP, Brasil

Abstract – In this work, we employed an unsupervised learning paradigm to understand how musical genres can be grouped. We worked with dimensionality reduction methods, such as PCA and UMAP, as well as clustering models like KMeans, Mean-Shift, and hierarchical clustering. Our results highlighted attributes such as musical mode, explicitness, energy, and acoustics as relevant for the clustering problem. Despite the significant variance in attributes within the same genre, we consistently found a similar grouping of genres across all clustering models. Thus, we foresee that our work could not only be a powerful tool for the quantified understanding of musical genres but also for data scientists seeking ways to preprocess their data for classification tasks.

Keywords – machine learning, clusterização

1. Introduction

Many Brazilians, including the renowned musician Zeca Pagodinho, attempt to clarify the differences between samba and pagode but fail to establish clear boundaries between these two musical genres [8]. On the other hand, it is intuitively easy to distinguish between genres like opera and grunge, as they exhibit opposite characteristics in some way. According to the musicologist Franco Fabbri, a musical genre can be understood as "a set of musical events whose course is governed by a defined set of socially accepted rules" [6]. Therefore, given that a genre can carry both real and subjective attributes, it is understandable that its distinction is not always clear.

On the other hand, with the advent of more computational resources, machine learning techniques, and extensive datasets, the automatic recognition of musical genres based on data science has become a growing field of study [7]. However, it still presents significant challenges, such as visualizing data from a vast number of musical genres and their similarities [12].

In this work, we used machine learning for clustering musical genres based on real attributes of songs from the Spotify™ music streaming program. To better explore the practical content of the IA048 course, going beyond what was covered in the practical activities, we used an unsupervised approach. Through both old and recent models, such as K-Means and Uniform Manifold Approximation and Projection (UMAP) [5], respectively, we were able to clearly identify the attributes of a song that make it belong to a certain genre and also provide clues to differentiate genres with similar attributes. Modestly, we believe that our work can contribute both to musicologists who wish to better understand the definition of musical genres and to inspire future machine learning students interested in the unsupervised paradigm.

2. Dataset description

In this work, we used Spotify™ music data made available on the Kaggle™ community [4]. The dataset contains 114 genres, and for each genre, we have a total of 1,000 songs. The attributes for each song are: track ID, artist, album name, popularity, duration (ms), explicitness (whether the lyrics contain explicit words), danceability, energy, key ($0 = C$, $2 = D$, and so on), average loudness (in dB), mode (whether the scale is minor or major), speechiness (presence of spoken words), acousticness, instrumentalness (predicts whether the track contains no vocals), liveness (presence of an audience), valence (detects the emotional positivity of the track), tempo (in beats per minute - BPM), time signature, and musical genre.

For the clustering task, we removed the attributes of track ID, artist, album name, and musical genre. The genre of the songs was later added for cluster comparison. Therefore, for each sample, we have 15 attributes that will be used to train the models. All these attributes were rescaled from 0 to 1.

3. Dimensionality reduction and data visualization

3.1. PCA

As an initial analysis of the data, we applied the Principal Component Analysis (PCA) method. It was identified that ten components explain 96.46% of the variance in the data, while two components account for only 45.21% of the variance. The first component has a strong negative correlation with the musical mode, and the sec-

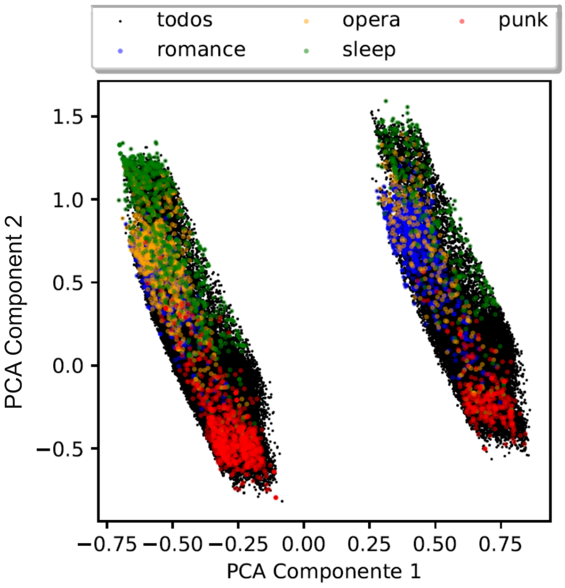

Figure 1. Distribution of genres in the space generated by the first two components of the PCA.

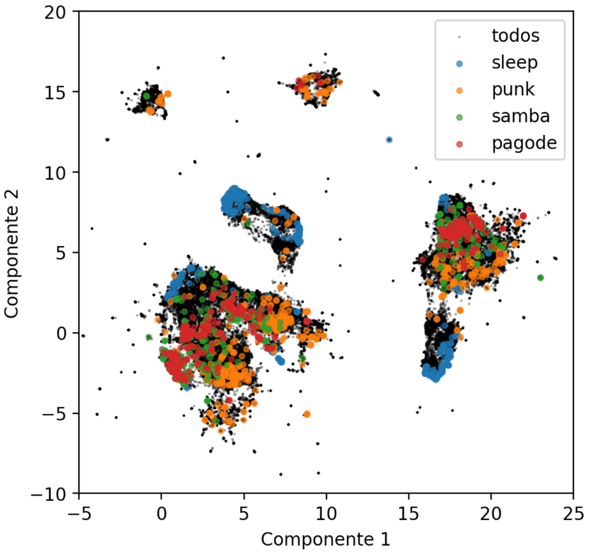

Figure 2. Visualization of music tracks based on genres through the UMAP model.

ond component has a positive correlation with acousticness and a negative correlation with the energy of the music. To illustrate how these components might relate to musical genres, Figure 1 shows the distribution of four different genres in the space generated by the first two PCA components. It is notable that the first component effectively separates the genres *romance* (blue) from *opera* (orange), as the former predominantly features a minor mode, while the latter features a major mode. The second component clearly separates the genres *sleep* (green) and *punk*, as their energy and acousticness are quite different.

Although we can infer some insights about the mentioned attributes, grouping them into principal components leads to a loss of intuition, making it harder to interpret the clusters. Therefore, we chose to use all 15 attributes as input for the model and not to use PCA for the clustering problem.

3.2. UMAP

Uniform Manifold Approximation and Projection (UMAP) is a dimensionality reduction method designed to facilitate data visualization in attribute space [5]. Similar to t-SNE (t-Distributed Stochastic Neighbor Embedding), it seeks to preserve the notion of neighborhood among samples but also ensures greater training efficiency compared to t-SNE. The model's hyperparameters include the number of neighbors (*n_neighbors*), the minimum allowable distance between points (*min_dist*), the number of components in the resulting space (*n_components*), and the type of metric used to compute distances (*metric*). Since we want to visualize the data in two dimensions, we used 2 for *n_components*. To simplify the hyperparameter search, we fixed *min_dist* at 0.1 and used the Euclidean metric. As this is primarily a data visualization technique, we employed human inference to determine the optimal *n_neighbors* for the model. We observed that the smaller the *n_neighbors*, the more clustered the points are, but after a certain point, specifically *n_neighbors* = 32, the data becomes well separated into different clusters, and the pattern does not change with more neighbors.

In Figure 2, we present the mapping performed using UMAP in two dimensions, where we highlight intuitively distant classes, such as *sleep* and *opera*, and closely related classes like *samba* and *pagode*. At first glance, the map is not able to separate musical genres well into distinct groups, as we can observe the presence of samples from the same class in almost all clusters, with the exception of the upper clusters, which are dominated by the *punk* genre. However, we can conclude some important aspects about the dataset. Although the separation of genres does not occur across different clusters, the genres *sleep* and *punk* are clearly opposite on the map, which aligns with our understanding since the energy levels of these two genres are completely different. Additionally, *pagode* appears to be contained within the space on the map occupied by *samba*, which could suggest that it might even be considered a subgenre of *samba*. Therefore, the UMAP model is able to separate musical genres in the reduced-dimensional space based solely on musical attributes. This suggests that the attributes may be sufficient to arrive at the notion of musical genres without needing to rely on historical aspects. A caveat, which is the subject of recent discussion, is that using UMAP for clustering tasks is quite controversial, as

UMAP or t-SNE do not fully preserve the density of samples, which could result in false clustering of the data [9]. Nevertheless, we demonstrate how UMAP is a powerful tool for making inferences that intuitively make sense about musical genres.

4. Clustering methods

4.1. Kmeans

To determine the appropriate number of clusters (*n_clusters*), we examined the Silhouette coefficient, defined as $S = \frac{b-a}{\max(a,b)}$, where a is the average intra-cluster distance and b is the average inter-cluster distance [10]. We initialized the cluster center positions using the "k-means++" method, which aims to maximize the probability that a candidate centroid point is far from the first already chosen cluster [2]. Figure 3a shows the average Silhouette coefficient S as a function of *n_clusters*, indicating better separation between the samples for *n_clusters*=2. However, when analyzing the distribution of attributes for 2 clusters, it was observed that the model relied solely on the *mode* attribute. To avoid triviality in the model's decision, we preferred to choose an *n_clusters* that still reasonably separates the data. We noticed that *n_clusters* = 5 seems to favor this balance, so we selected this value for our analyses. In Figure 3b, we conducted a Silhouette analysis for *n_clusters* = 5, which shows the S values for each sample, with an overall average of $S = 0.23$, implying that most samples are close to the decision boundaries. Additionally, we can see that cluster 4 does not separate well from the other samples, while clusters 0, 1, and 2 contain the majority of the samples.

4.2. Mean-shift

In the Mean-Shift (MS) algorithm, a kernel function is used to calculate the weights of a weighted average of the [3] samples. In this work we use the Gaussian function, and the only hyperparameter to be optimized is the bandwidth of the Gaussian function. We trained the model with different bandwidths and discovered that this hyperparameter directly influences the number of clusters, as demonstrated in figure 4. It is expected that the greater the bandwidth, the smaller the number of clusters calculated, as the Gaussian function for calculating the sample mean will be wider. To maintain the standard between models, we set a desired number of clusters of 5, which implies a bandwidth of 1.

4.3. Hierarchical clustering

For hierarchical clustering, we use the *cluster.hierarchy* function from the *scipy* library. As an optimization method, we use Ward variance [11]. The dendrogram generated by hierarchical clustering on the input data is shown in

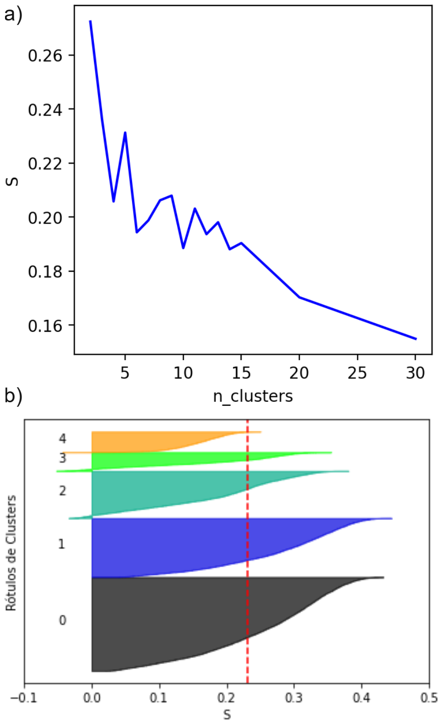

Figure 3. a) Silhouette Coefficient depending on the number of clusters. b) Silhouette Analysis. The red dashed line indicates the global average of the S coefficient.

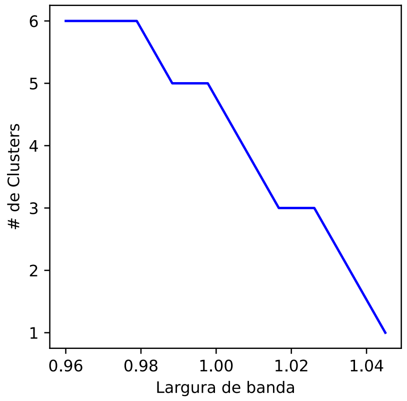

Figure 4. Number of clusters as a function of the bandwidth of the Gaussian function used in the MS method.

figure **??**. To have more details on the internal groupings of each branch of the dendrogram, we set a distance value of *threshold* at 28, thus resulting in 6 clusters. The distribution of attributes for each cluster will be commented in the results section.

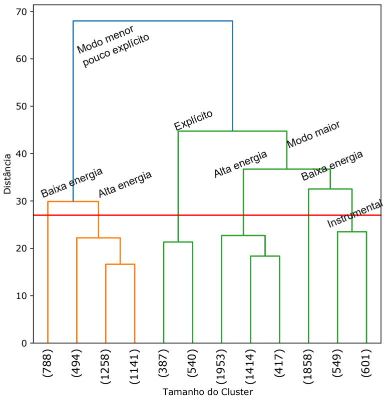

Figure 5. Dendrogram generated by hierarchical clustering. For each branch, we highlight the main attributes. The red line represents the distance cut performed to obtain 6 clusters.

4.4. Other methods

In addition to these clustering methods mentioned, there are other well-established models, such as the Kohonen map, spectral clustering, and affinity propagation. In this section we expose our experiences with each of these models applied to the music clustering problem. One of the difficulties we encountered when applying the Kohonen map is the lack of robustness in the ready-made libraries available. Several users reported similar problems, such as the inability to view the network weights and the lack of a standard in the sklearn [1] API. The affinity propagation algorithm suffers in the training stage as it requires more RAM memory than that provided free of charge by Google Colab™Ültimately, spectral clustering worked well, however, it generated very similar results to KMeans. To maintain the brevity of this report, we decided not to present them here.

5. Results and Discussion

To evaluate the performance of each model, we investigated the distribution of attributes for each cluster, and exemplified some of the predominant genres for them. The results are presented in the tables 1, 2, and 3, for the KMeans, MS, and hierarchical clustering models, respectively.

In general, the most important attributes for separating clusters were: the mode, energy and explicitness of the music, as expected by the PCA. For the Kmeans algorithm, we realized that the simple difference in the mode (minor or major) of the music was enough to separate genres like pop and party, from genres like *house-music* and its subvariants. Furthermore, it is surprising that within the same cluster (2 - KMeans), we find genres such as *honky-tonky* (reminiscent of a fast piano that played in bars in the old American West), and classical songs and operas. But when observing the distribution of attributes in this cluster, it becomes clear that what unites these genres is that the majority of them are in major mode, are acoustic and are generally of lower volume. Genres such as romance and tango are also well separated from the rest, as they are low in energy, are in minor mode and are acoustic. Interestingly, the genres of music that were closest to the centroids for clusters 0 to 5 were: *blues*, *children*, *children*, *cantopop* and *blackmetal* (track indexes for reader consultation: [8912, 14209, 14042, 12847, 6217]). Therefore, although there are predominant genres for each cluster, the centroids do not determine the type of musical genre. This is reasonable since within the same genre, it is possible to find a large variance in attributes.

The distribution of attributes for the clusters generated by Mean-Shift is very similar to KMeans, containing a cluster with high musical energy, a second cluster with *house-music* and one with explicit words whose predominant musical genre is *comedy*. The difference between the models is mainly in clusters 2 and 3 of KMeans and 3 and 4 of Mean-Shift. For example, because Mean-Shift weighed more on the explicitness of the music, it was able to better identify genres of the *ambient* type, but on the other hand it placed the *romance* genre in the *house-music* for presenting a minor musical mode.

Finally, hierarchical clustering allows us to create a kind of decision tree based on musical attributes (see figure **??**). The main attributes for the first separation of branches were the musical mode and explicitness. On the left side of the dendrogram, there are less explicit songs with a minor mode, such as *romance* and *tango* (low energy), and *house-music* (high energy). It is notable that this branch is similar to KMeans clusters 3 and 1, respectively. The second branch is separated into 4 smaller clusters based on the explicitness, energy, and instrumentality of the songs in hierarchical order of distances. Clusters IIb2a (little instrumental) and IIb2b (very instrumental) are very similar to the clusters with KMeans cluster 2 and Mean-Shift cluster 4. This means that like other clustering models, hierarchical clustering can capture the essence of some genres based only on musical attributes. Furthermore, this type of clustering offers a possible organization of clusters, although this tree-like structure is not related to historical aspects.

Cluster Label	Relevant Features	Predominant Genres
0	danceable, energetic, major key	power-pop, party, j-idol, ska
1	danceable, energetic, minor key	drum-and-bass, progressive-house...
2	energetic, low volume, major key, acoustic	opera, classical, honky-tonk
3	low energy, low volume, minor key, acoustic	romance, tango
4	explicit, danceable, energetic, spoken, low instrumental	comedy

Table 1. Distribution of features for each group generated by the KMeans method.

Cluster Label	Relevant Features	Predominant Genres
0	Not explicit, energetic, major key	country, songwriter, opera, disney
1	Not explicit, energetic, minor key	romance, trip-house, deep-house
2	Explicit, energetic, major key	comedy
3	Explicit, energetic, minor key	j-dance
4	Not explicit, low energy, low volume, major key, acoustic	ambient

Table 2. Distribution of features for each group generated by the Mean-Shift method.

Cluster	Cluster Characteristics	Predominant Genres
Ia	Minor key, not explicit, low energy	romance, tango
Ib	Minor key, not explicit, high energy	deep-house, drum-and-bass, trip-hop
IIa	Explicit	sad
IIb1	Not explicit, major key, low instrumental, high energy, low acoustic	party, comedy, country, punk
IIb2a	Not explicit, major key, low instrumental	opera, honky-tonk, songwriter
IIb2b	Not explicit, major key, high instrumental	ambient, study, classical

Table 3. Distribution of features for each group generated by hierarchical clustering.

6. Conclusions

Briefly, our work utilized the unsupervised paradigm to understand how songs can be grouped and how these clusters resemble musical genres. By exploring dimensionality reduction methods like PCA and UMAP, it was possible to gain insights into the important attributes for sample separation, such as mode, energy, and musical acoustics. Additionally, through the visualization of samples in reduced dimensions, we concluded that some genres are easier to separate from one another, such as *sleep* and *punk*, while others are a bit more challenging, like *samba* and *pagode*, given that the latter two are very similar. However, even similar genres like these show different variance in the reduced dimensional space, which may imply that one genre is contained within the other, but for historical reasons, they are considered different genres.

Moreover, among the three clustering methods applied in this work—KMeans, Mean-Shift, and hierarchical clustering—the latter model was the most informative, as it allows for an organization of attribute structures similar to a decision tree. Nonetheless, the KMeans and Mean-Shift methods also proved effective in grouping songs with similar attributes that clearly have predominant genres. One of the most notable cases was the *house music* genre, which, due to its unique characteristics, such as minor key, low explicitness, and high energy, was separated from other genres by all three clustering methods presented. On the other hand, we observed subtle differences between methods, such as in the case of the *romance* genre, which was grouped differently depending on the model used. Some limitations of the clustering problem applied to this dataset can be highlighted, such as the difficulty in separating very similar genres and the challenge of dealing with the immense variance of attributes within a single genre. However, we emphasize that the findings made here can aid in the preprocessing of datasets prepared for the genre classification problem, which consequently has immediate applications such as automatic and personalized playlists for music streaming services.

Acknowledge

The authors would like to thank professors Levy Boccato and Romis Attux for class discussions in the machine learning discipline. We also thank FAPESP (2020/15740-3) for financial support.

References

[1] sklearn-som 1.1.0. `https://pypi.org/project/sklearn-som/`. (acessado em

08/12/2022).

[2] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[3] Jianan Lin. Understanding mean shift clustering and implementation with python. `https://towardsdatascience.com/`. (acessado em 08/12/2022).

[4] MAHARSHIPANDYA. Spotify tracks dataset. `https://www.kaggle.com/`. (acessado em 02/11/2022).

[5] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[6] Allan F Moore. Categorical conventions in music discourse: Style and genre. *Music & letters*, 82(3):432–442, 2001.

[7] François Pachet, Gert Westermann, and Damien Laigre. Musical data mining for electronic music distribution. In *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*, pages 101–106. IEEE, 2001.

[8] Zéca Pagodinho. Zeca pagodinho "É igual mais é diferente" programa do jô soares. `https://www.youtube.com/watch?v=7lb5IIndy8Y`. (acessado em 02/12/2022).

[9] Leland McInnes Revision. Using umap for clustering. `https://umap-learn.readthedocs.io/`. (acessado em 07/12/2022).

[10] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[11] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[12] Janice Wong. Visualising music: the problems with genre classification. `https://mastersofmedia.hum.uva.nl/`. (acessado em 02/12/2022).