Machine Learning Tasks on Musical Data

Britt Lange

1 Introduction

Music is a powerfully unique art form; while deeply subjective, as music can provoke intense feelings in each listener, the principles of Music Theory provide virtually-mathematical formulas that allow us to methodically describe a large portion of such an abstract medium. But to what extent can music be quantified? Modern digital music databases such as Spotify have developed algorithms that can analyze the mood or danceability of a user's listening history or provide personalized recommendations of similar playlists, artists, and songs to a user's favorites. How are these calculations made, and how are the audio features of music extracted so as to offer favorable results? In this paper I will explore modern methods of Music Information Retrieval (MIR) and analyze the use of different forms of musical data for machine learning tasks.

2 Music Information Retrieval

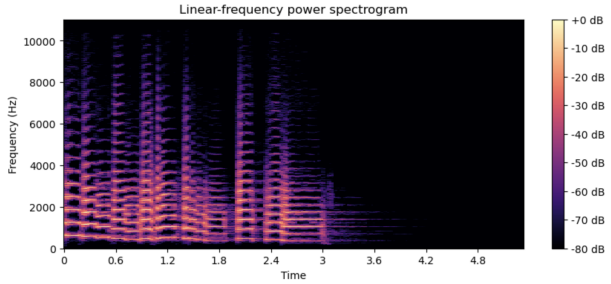
Music Information Retrieval (MIR) is a field that confronts the challenge of representing music as quantitative data and automatically extracting audio features from musical data. Downie (2003) mentions "music's pitch, temporal, harmonic, timbral, editorial, textual, and bibliographic 'facets'" as the primary representational aims of MIR, noting the difficulty in capturing each of these aspects. In addition to these challenges, Downie enumerates three overarching challenges of MIR:

- **The Multirepresentational Challenge:** Music can be represented as audio data or as textual data with symbolic notation. Symbolic musical notation requires less computational bandwidth to compute as data, although most consumers of music view it as a primarily auditory medium. Converting audio data into quantitative data requires expensive computational resources.
- **The Multicultural Challenge:** In the MIR realm, a large majority of the audio data that has been collected and analyzed adheres to the practices of Western classical or popular music. Therefore, in describing a piece of audio data, Western Common Practice (CP) notation may be the only framework available, whereas the music involved may lie far outside the descriptive abilities of this framework. With many musical styles and cultures unavailable in digital form, the MIR realm lacks the resources necessary to describe a significant portion of existing music.
- **The Multiexperiential Challenge:** Music is extremely subjective. The experience of a given piece of music varies widely between listeners, and there is no one "correct" use of any piece of music. As Downie notes, music can be experienced through live performances like concerts, in other expressive art forms like dance, in religious practices, or as background noise to everyday tasks. In describing the experience of any piece of music, there will always be an audience who is addressed by such descriptions and an audience who is neglected.

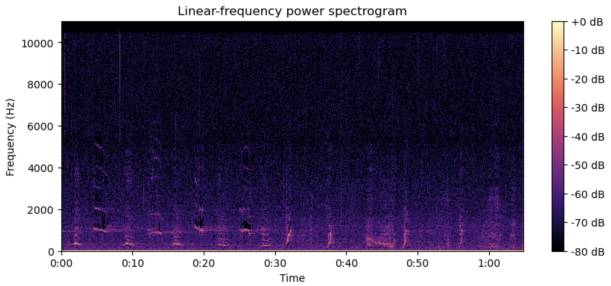
Given music's breadth and subjectivity, though textual metadata (including artist, title, album, genre, and release year) can easily be retrieved and manipulated as data, content-based MIR tasks require advanced audio processing capabilities.

2.1 Spectrograms

Many modern computer tasks performed on music first convert audio data into visual representations called spectrograms. A spectrogram consists of an x-axis representing time, a y-axis representing audio frequency, and different colors representing the amplitude of a particular frequency at a given



(a) Spectrogram of librosa audio example of monophonic trumpet sequence.



(b) Spectrogram of librosa audio example of 60-second humpback whale song.

time. Figure 1a represents the spectrogram of a five-second monophonic trumpet sequence, contrasted with Figure 1b's spectrogram representation of a 60-second humpback whale song.

While several machine learning methods have been developed for tasks such as musical instrument differentiation, timbre identification, and music genre classification through spectrograms, some low-level automatic feature extraction from audio data can be performed with the Python library "librosa". Librosa provides functions that can convert an audio file into a spectrogram, return the most likely musical notes based on the frequency of a given audio file, and analyze the rhythm of a spectrogram (McFee et al., 2015). What cannot be computed with this library is timbre, texture, or harmonic analysis — the more subjective elements of music. Much more advanced computer techniques are required for such high-level musical analysis.

3 User Feedback Machine Learning Algorithms for Music Recommendation

Many algorithms used by music platforms like Spotify provide users with high-level music recommendations that do not require audio features; instead, they rely on metadata or user activity.

3.1 Collaborative Filtering for Music Recommendation

Collaborative filtering (CF) refers to the use of matrices containing users' ratings for various items in a dataset in order to calculate between-item or between-user similarity, and it has been widely used to perform music recommendation tasks (Vasu and Choudhary, 2022; Sánchez-Moreno et al., 2016). While it can be difficult to collect datasets involving user ratings, implicit ratings can be derived from user behavior on platforms like Spotify (Sánchez-Moreno et al., 2016). There are two different methods of collaborative filtering: memory-based and model-based.

Memory-based CF involves user-item comparisons calculated with the nearest-neighbor method. A user (listener) rates an item (song, playlist, etc.) with a particular score depending on how they interact with the item, and nearest neighbor methods scan for users who have rated common items with similar scores. While easy to implement with the most up-to-date datasets, memory-based CF poses a scalability problem: with larger datasets, scanning the entire dataset for neighbors becomes slow and expensive (Sánchez-Moreno et al., 2016).

Model-based CF addresses this scalability problem by pre-computing item-item similarity given existing user ratings. Whereas memory-based CF computes all similarities at recommendation time, model-based CF can continually predict new item ratings in the similarity matrix before a recommendation is needed. This computation utilizes Machine Learning models such as Clustering and Matrix Factorization, and similarity is often represented by cosine similarity or, less frequently, the Pearson correlation coefficient (Sánchez-Moreno et al., 2016). Because computations can be performed before recommendation time, model-based CF does not face the scalability problem, though results may not be as accurate as those of memory-based CF.

CF models can be modified to better suit music recommendation tasks; Sánchez-Moreno et al. (2016) propose a model that, in addition to computing item-item similarity, predicts item popularity based on volume of users who interact with the item and number of interactions the item has. They determine a listening coefficient for artist popularity given this user data, and this artist popularity coefficient is used to determine a user coefficient that characterizes users based on the popularity of

the artists they listen to. They incorporate these coefficients into model-based CF recommendation systems, adding another dimension to the cosine similarity computed between users and items and providing higher-level recommendations.

CF-based methods are effective in recommending music between similar users and items. But there are several challenges with user-based data that CF cannot sufficiently confront: the problem of early-rater items, cold-start users, and gray-sheep users.

Early-rater items refer to items that are new to a database and have not been rated by enough users to be able to compute meaningful similarities or select the item for recommendations. Similarly, cold-start users are users who are new to a platform and have not rated enough items to receive effective recommendations. Gray sheep users have preferences that are too unique to receive relevant recommendations. To successfully overcome these challenges in user-item algorithms, incorporation of features of the content itself is necessary.

4 Content-Based Machine Learning Algorithms for Music Analysis and Classification

As seen with user-feedback-based algorithms like Collaborative Filtering, issues arise when items are introduced to a system without initial user ratings or users enter a system without having rated any items. In these cases, content-based techniques must be incorporated to fill gaps in user-based data.

4.1 Clustering of Echo Nest Features for Playlist Analysis

Pichl et al. (2017) explore several content-based methods for analyzing user-curated playlists on Spotify. Using an Echo Nest audio summary for the tracks in their dataset, they first perform a Principal Component Analysis (PCA), which involves the data's correlations and covariance matrices: correlations between features are analyzed, and features that produce the most variation in the data are identified as the principal components.

They then use distance-based k-Means clustering to aggregate playlists into groups, estimating k with the results of the PCA. Additionally, they make use of graph-based spectral clustering to represent non-linear relationships in the data; this method translates connectivity between data points to an adjacency matrix which is then projected into a lower-dimensional space in which clustering can be applied.

Their analyses of the different audio features of playlist clusters can be useful for music recommendation platforms like Spotify to create and recommend user-oriented playlists. Incorporating the user feedback techniques of collaborative filtering, this audio feature analysis could be useful in generating user-specific playlists and recommendations.

5 Support Vector Machines for Spectrogram Genre Classification

(Costa et al., 2011) holds music genre to be the "most widely used" descriptor of music to "organize and manage large digital music databases", and they propose the use of spectrograms to locally extract visual features and perform more robust classification tasks.

To extract textural features from the spectrogram, they use Gray-Level Co-Occurrence Matrix (GLCM) descriptors, which "provides measures of properties such as smoothness, coarseness, and regularity" (Costa et al., 2011). Utilizing a subset of GLCM features and a zoning mechanism to obtain local information, they classify music genre using a Support Vector Machine (SVM).

Performing experiments on the Latin Music Database and using 3-fold cross validation, they compare their classification model to a previously implemented model that used "feature vectors representing short-term, low-level characteristics of music audio signals" (Costa et al., 2011). The spectrogram-based model performed more strongly than the baseline by a difference in average recognition rate of 0.5%, though for many genres both models performed quite poorly. The overall recognition rate of the spectrogram-based model was 60.1%, but finally they summed the two classifiers, achieving a combined recognition rate of 67.2%. Their experiments display how texture images such as spectrograms can be used as input to classification tasks, complementing information provided by short-term, low-level characteristics of audio signals when models are combined.

5.1 Convolutional Neural Networks for Spectrogram Genre Classification

Costa et al. (2017) build and evaluate the performance of a Convolutional Neural Network (CNN) for music genre classification from spectrograms,

comparing their model's use of representation learning for feature extraction to models using hand-crafted feature extractions. They hypothesize that representation learning from visual data representations is more effective than handcrafted features taken directly from audio signals or time-frequency images.

Discussing the architecture of their CNN, they do not utilize domain knowledge and instead allow the model learn the design of feature extractors, and they found square-shaped kernels to be the most effective in capturing the features.

They compared their model's classification to that of Support Vector Machines (SVM) applied to the handcrafted audio features, particularly to data representations through Robust Local Binary Patterns (RLBP). LBP is used to represent local visual features with binary strings based on a central pixel and its circular surroundings, and RLBP is an extension of LBP that is more effective in identifying and treating noise (Chen et al., 2013). The other models tested used hand-crafted acoustic features, and the CNN and RLBP were the only models employing visual data representations.

Training and testing these models on several datasets, they found that CNNs outperformed all other models in two of three cases, with RLBPs prevailing on one dataset. But they also built a fusion of these two models for each dataset, combining the two classifiers. The best results for each dataset were produced when RLBPs were fused with CNNs, suggesting the strength of visual data over acoustic data. Their experimentation also provides a compelling case for the effectiveness of representation learning over hand-crafted feature engineering for music content classification.

5.2 Convolutional Neural Networks for Spectrogram Timbre Classification

Various Machine Learning tasks have been performed on music spectrograms for genre and mood classification (Álvarez et al., 2022; Dalida et al., 2022; Costa et al., 2011, 2017), and Spotify's Echo Nest API has become standard for music emotion classification. Classification of musical timbre, however, is a task that is much less frequently attempted and involves more complicated Machine Learning architectures.

Pons et al. (2017) attempts to use a Convolutional Neural Network to identify timbre of music using spectrograms. Timbre is one of the most subjective elements of music; it refers to the quality of the sound of a piece of music and varies between instruments, textures, and pitches. Examples of timbre identifiers might be "airy", "full", or "hollow". Timbre is a unique aspect of music that seems to require a human-level understanding of sound.

The researchers in this experiment provide thorough descriptions of their careful choices for the sizes and shapes of their kernels, or filters, and the layers of the neural networks. As they point out, there are risks involved with choosing the filter shapes; on one hand, too much noise might be captured with a larger filter, though on the other hand, a smaller filter might not capture the full context of a sound. For these reasons, they assert that using several differently shaped filters in the first layer is necessary to mitigate these risks.

They perform several classification tasks related to timbre. The first is singing phoneme classification, which involves identifying the shape of each vocal fragment in a jingu (Beijing opera) a cappella audio dataset. With 32 phoneme classes and two separate datasets consisting of 89 and 39 minutes of audio data, they build their neural network for this task using "128 filters of sizes 50×1 and 70×1, 64 filters of sizes 50×5 and 70×5, and 32 filters of sizes 50×10 and 70×10," given the complications involved with precisely capturing each audio fragment (Pons et al., 2017). Using one wide convolutional layer including these filters, they compare the accuracy of their model to that of three baseline models on the same datasets. Their model performs more strongly than the other models, with a 0.484 and 0.432 accuracy on both datasets and the closest contender being a 5-layer CNN with small-rectangular filters of size 3x3, which achieved 0.374 and 0.359 accuracy on the same datasets, respectively. The researchers point out that this CNN with small-rectangular filters is state-of-the-art on larger datasets, but does not perform as well on this smaller dataset.

The second timbre-related task they perform is musical instrument identification. For this task, they build two neural networks: one with a single convolutional layer and one with multiple convolutional layers. They compare both of their models to two baselines, one being state-of-the-art for this task. With 6705 3-second audio excerpts, each involving one primary instrument, they train and test their models against the baselines on this dataset. The accuracy of their multi-layer model achieves near-state-of-the-art accuracy on several

tests, reaching around 50—60%.

With meticulously fine-tuned neural networks optimized to fit each task, the performance of these models on timbre-related tasks, especially the singing phoneme classification task, does not produce very effective results. While the researchers design their models carefully to improve upon previous models, progress remains slow and gradual. Performance of these models are improving, little by little, but at the moment, they are still not reliable enough to produce credible results.

6 Conclusions and Implications

These experiments pose the question, how much of the experience of music can be captured through computerized musical data? Music is a unique art form in that it has its own language in the form of musical notation; pitches, intervals, and harmonic progressions are assigned values that are represented as numerals. While mathematical formulas and symbols can describe how a piece of music ought to be played, in order to holistically understand a piece of music, one must hear it.

Spectrograms are visual representations of audio files that allow the shapes, pitches, and volume of sounds to be analyzed. But by converting audio data into visual data, what aspects of true sound are lost? One can study the shapes and formulas involved in a piece of music and gain a complete understanding of how it should sound, but how can anyone, human or machine, hear music without actually hearing it? Different musicians with the same instrument might play the same piece with completely different styles or tones. An audio recording transformed into a spectrogram might hold the necessary information to understand sound, but this requires effective Machine Learning methods of extracting the most subtle and subjective features.

It seems as though these algorithms are slowly improving in their recognition of data that, in its rawest form, is not quantitative. But for now, it seems that music production and deep analysis remains a human art form.

References

Jie Chen, Vili Kellokumpu, Guoying Zhao, and Matti Pietikäinen. 2013. Rlbp: Robust local binary pattern. In *British Machine Vision Conference*.

Yandre Costa, Luiz Soares de Oliveira, A.L. Koericb, and Fabien Gouyon. 2011. Music genre recognition using spectrograms. pages 1 – 4.

Yandre M.G. Costa, Luiz S. Oliveira, and Carlos N. Silla Jr. 2017. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied Soft Computing*, 52:28–38.

Marvin Ray Dalida, Lyah Bianca Aquino, William Cris Hod, Rachelle Ann Agapor, Shekinah Huyo-a, and Gabriel Sampedro. 2022. Music mood prediction based on spotify's audio features using logistic regression. pages 1–5.

J. Stephen Downie. 2003. Music information retrieval. In *Annual Review of Information Science and Technology*, chapter 7, pages 295–340.

Brian McFee, Collin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python.

Martin Pichl, Eva Zangerle, and Günther Specht. 2017. Understanding user-curated playlists on spotify: A machine learning approach. *International Journal of Multimedia Data Engineering and Management*, 8:44–59.

Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. 2017. Timbre analysis of music audio signals with convolutional neural networks. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2744–2748.

Diego Sánchez-Moreno, Ana B. Gil González, M. Dolores Muñoz Vicente, Vivian F. López Batista, and María N. Moreno García. 2016. A collaborative filtering method for music recommendation using playing coefficients for artists and users. *Expert Systems With Applications*, 66:234–244.

Karthik Vasu and Savita Choudhary. 2022. Music information retrieval using similarity based relevance ranking techniques. *Scalable Computing: Practice and Experience*, 23.

P. Álvarez, J. García de Quirós, and S. Baldassarri. 2022. Riada: A machine-learning based infrastructure for recognising the emotions of spotify songs. *IJIMAI*.