Graph Neural Networks for Music Genre Classification

Shardul Dhongade
Virginia Tech
shardul21@vt.edu

Nikolas Rovira
Virginia Tech
nikolasr20@vt.edu

Abstract

Music genre classification is a well-explored application of deep learning, often employing Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to identify patterns in audio features. With the growing popularity of Graph Neural Networks (GNNs) and their emerging applications, this paper explores the use of GNNs for music genre classification. Prior work has been done to create music recommender systems with GNN's, but not in-depth. Building off some ideas from recommenders, we will build a classification model. We propose an approach leveraging GNNs to model the relationships between musical features and genres, and compare its performance against standard methods such as CNNs and RNNs. Experimental results will aim to demonstrate the potential advantages and limitations of GNN-based models in this context.

Keywords

Graph, Neural Networks, Music Genre, Classification

ACM Reference Format:
Shardul Dhongade and Nikolas Rovira. 2018. Graph Neural Networks for Music Genre Classification. In . ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Music genre classification is a common project done within Deep Learning, with further applications ranging from music recommendation systems to personalized playlist generation. Classification operates by categorizing music tracks into predefined genres based on the given song's audio features and patterns. Traditional machine learning approaches for audio tasks rely initially on feature extraction, where domain-specific features such as Mel-Frequency Cepstral Coefficients (MFCC) and chroma features [3] are selected and preprocessed, then fed into the model. The most commonly employed deep learning techniques for music genre classification include Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs excel with audio spectrograms [1], and RNN's have shown promise with Long Short Term Memory Network (LSTM) [4]. Both approaches have shown promising performance in several studies.

While CNNs and RNNs have been proven to be successful, they lack the degree of flexibility that GNNs may offer. Neural Networks can be limited when capturing complex relationships in data elements, such as the interactions between different musical features. GNNs provide a more flexible framework to model data with graph structures. GNNs have already proven effective in various domains like social networks, molecular chemistry [5], and natural language processing, where data can be represented as nodes and edges.

In this paper, we propose an approach to music genre classification using Graph Neural Networks. Prior work in this field discusses the use of GNNs for music recommender systems [9]. We aim to explore the effectiveness of GNNs in capturing relationships between musical features and genres, and compare their performance against commonly used CNN and RNN-based methods. The goal is to add contributions alongside current work on whether GNNs can offer a novel perspective and improved results in the domain of music genre classification.

2 Related Work

2.1 GNNs for Music Recommendations

Given the large amount of music data as well as the economic incentive for the efficient supply of music, a large amount of research has been performed focusing on music processing and recommendation. One example of this is Ben Alexander, Jean-Peic Chou, and Aman Bansal's *Implement Your Own Music Recommender with Graph Neural Networks (LightGCN)* [9]. This research used data from the Spotify Million Playlist Dataset to train a GNN to recommend songs to be added to playlists. They utilized the playlists and the songs within them to suggest tracks for playlists considered similar. This nature of the dataset indicates that the music recommendations are based on user behavior. If multiple users listen to the same songs in the same playlists, the suggested songs will reflect those patterns.

2.2 Link Prediction

With the rising popularity of GNNs, more means of training and optimizing them have been developed. One pertinent example of this is link prediction. In the paper *Link Prediction Based on Graph Neural Networks* [10] by Muhan Zhang and Yixin Chen, a link prediction framework named SEAL (learning from Subgraphs, Embeddings and Attributes for Link prediction) is proposed to perform link prediction. This method uses node distances and also injects new positive and negative edges into the graph to generate a new set of node embeddings a new subgraph. The unique operation performed in the SEAL method is the use of negative training links, which is a process they named *negative injection*. The inclusion of these negative training links prevents the GNN from only fitting to the positive links and subsequently generalizing poorly.

Link prediction is also addressed in *Modeling Relational Data with Graph Convolutional Networks* [11]. Here, the authors use an auto-encoder comprised of an entity encoder and a scoring function, operating as a decoder, to assign scores to possible edges

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

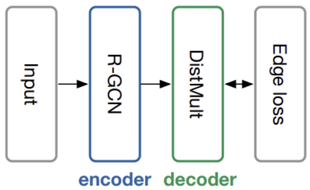

Figure 1: Link prediction model with an R-GCN encoder (interspersed with fully connected/dense layers) and a DistMult decoder that takes pairs of hidden node representations and produces a score for every (potential) edge in the graph. The loss is evaluated per edge.[11]

(seen in Figure 1). This use of an encoder, an R-GCN encoder in this case, differentiates it from other applications. The encoder maps each vertex to a real valued vector based on its features and relationships. The decoder takes the vertex representations provided by the encoder and reconstructs the edges of the graph.

3 Proposed Contributions

Our proposed contributions will be to borrow current practices with GNNs for recommender systems, and tweak them to be built for a classification model. This will include changing how data in each node for the graph is represented, whether it be as MFCCs or other. It will also comprise of reevaluating the message passing function for the edges of the graph. We will utilize the PyG library for the graph and build upon the current layers of the network. We will also add our own evaluation metrics to determine the model performance compared to industry accepted models using CNNs and RNNs.

4 Evaluation Plan

4.1 Datasets

The main datasets we are looking at will be the GTZAN audio dataset [8], which has 1000 tracks each of 30 second length. This is a commonly used dataset for music domain work, which will simplify the feature extraction process. Depending on our progress and feature extraction difficulties, we may also use the Million Song Dataset [9], which instead contains the audio features and metadata on one million songs. Music recommender systems have used this dataset in prior work as well, which may be of use.

4.2 Preparing the Data

Initial work with the data will include preprocessing the audio files using the LibRosa Library for music and audio analysis.

This library will provide the necessary means to extract the key features discussed, such as MFCC, and present the spectrograms representing the MFCC's. The MEL-spectrogram will create the MFCC, for a numerical representation. Prior preprocessing work has been done in the past. If we decide to borrow previously completed feature extraction work that meets our requirements, it will be cited.

Feature extraction using CNNs has also been done [6], and this process may be used as an alternative.

Once done, the MFCC values for each audio data will be created. This will be presented in vector representation, and passed as nodes for the graph.

4.3 Building the Model

There are a few steps involved in the GNN architecture. Each layer generally consists of a message function for passing information, aggregation for producing values, and updating the previous layers node. The goal will be to have the edges between nodes defined on the similarity of the respective node's features. Then the information with the message function can be propagated throughout the nodes. To assist with message passing and propagation, the Pytorch PyG library will be utilized, which is popular for GNN models.

From here, we will need to conduct more research on how to effectively train the model and creating the propagation layers. One idea we may use is the LightGCN [7], which is useful to conduct filtering and tweaking its embeddings.

4.4 Link Prediction

Link prediction will be performed based on the methodology provided in *Modeling Relational Data with Graph Convolutional Networks*, which was addressed in section 2.2. This methodology has already been implemented in the PyG library noted in section 4.3. The implementation would only need to be changed to be used on our music datasets as opposed to the FB15k237 datasets.

5 Primary Experiments

Primary work will consist of determining the optimal way to represent the nodes in the graph, whether it be as MFCCs or vectored MEL spectrograms. We have started looking into building the graph nodes and their representations. No further experiments have been conducted yet, as we are still taking time to understand how the graph and how the propagation step will be conducted. However, we do have our next steps planned out. This will mainly include taking the graph nodes and building out the message function for the edges, to propagate information for updating the node [12].

6 Work Distribution

The work will be split evenly amongst the two of us. With both of us being new to deep learning and this topic, we will work alongside each other in most areas.

Shardul will focus on the domain-specific concepts regarding audio data and preprocessing. He will also assist on building the GNN and its architecture, propagation, as well as with training the model.

Nikolas will focus on testing and evaluation metrics for the model. He will also assist on building the GNN, propagation, and training the model.

Overall, both members will work cooperatively on several areas.

References

[1] Reddy, A. (2023) Convolutional Neural Networks (cnns) for audio data or audio feature extraction, Medium. https://anudeepareddy-s.medium.com/convolutional-neural-networks-cnns-for-audio-data-or-audio-feature-extraction-1c41f4aac35d (Accessed: 21 October 2024).

[2] Noopur Srivastava, Shivan Ruhil, Gaurav Kaushal. Music Genre Classification using Convolutional Recurrent Neural Networks, 2022 IEEE 6th Conference on Information and Communication Technology (CICT) [Preprint]. doi:10.1109/cict56698.2022.9997961.

[3] Julian Urbano, Dmitry Bogdanov, Perfecto Herrera, Emilia Gomez and Xavier Serra. "What is the effect of audio quality on the robustness of MFCCs and chroma features?". Proceedings of the 15th Conference of the International Society for Music Information Retrieval (ISMIR 2014); 2014 Oct 27-31; Taipei, Taiwan. International Society for Music Information Retrieval; 2014. p. 573-78.

[4] Prem Tibadiya. 2020. Music Genre Classification using RNN-LSTM. (July 2020). Retrieved October 21, 2024 from https://medium.com/@premtibadiya/music-genre-classification-using-rnn-lstm-1c212ba21e06.

[5] Yuyang Wang, Zijie Li, and Amir Barati Farimani. Graph Neural Networks for Molecules. A Chapter for Book “Machine Learning in Molecular Sciences". arXiv:2209.05582v2 (2023).

[6] Manoranjan Sathyamurthy, Xiaowen Dong, and M. Pawan Kumar. "Geometric Deep Learning for Music Genre Classification". MIT Media Lab

[7] He, Xiangnan, et al. “LightGCN: Simplifying and powering graph convolution network for recommendation.” Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020.

[8] Bob L. Sturm. (2013) The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. arXiv:1306.1461v2

[9] Alexander, B. (2021) Implement your own music recommender with Graph Neural Networks (LightGCN), Medium. Available at: https://medium.com/@benalex/implement-your-own-music-recommender-with-graph-neural-networks-lightgcn-f59e3bf5f8f5 (Accessed: 21 October 2024).

[10]] Muhan Zhang, Yixin Chen. Link Prediction Based on Graph Neural Networks. (November 2018). Cornell University. arXiv:1802.09691

[11] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, Max Welling. Modeling Relation Data with Graph Convolutional Networks. (March 2017). Cornell University. arXiv:1703.06103

[12] Emma Benjaminson. "Understanding Message Passing in GNNs". https://sassafras13.github.io/GNN/.