MSc in Artificial Intelligence

NCSR Demokritos & University of Piraeus

**Song similarity retrieval**

**Multimodal Machine Learning**

**Students**: Yiannis Savvas, Chaido Poulianou

Εικόνα που περιέχει κείμενο, γραμματοσειρά, γραμμή, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Sections

[1 Introduction 1](#_Toc170600784)

[2 Architecture pipeline 2](#_Toc170600785)

[2.1 Datasets 3](#_Toc170600786)

[2.2 Classifier models 3](#_Toc170600787)

[2.3 Metadata 5](#_Toc170600788)

[2.4 Autoencoder 5](#_Toc170600789)

[3. Similarity 6](#_Toc170600790)

[4. Experiments 6](#_Toc170600791)

[5. Conclusion 9](#_Toc170600792)

[6. References 10](#_Toc170600793)

## 1 Introduction

In this project, we deal with the issue of a song recommendation system based on closest similarity metric comparison to a song database. Our goal is to find a well-represented mapping of the different song characteristics that depend on song similarity to a latent space. Among the many different approaches for this problem like collaborative filtering, we have decided to utilize representation learning with two modalities: the song’s raw audio and text metadata. To further enrich and refine our representations, we use both supervised and unsupervised feature embeddings with the use of a CNN and Autoencoder architecture.

We explain our data preprocessing in section 2, where we used audio feature extraction to extract the Mel-spectrograms of the training dataset songs. In section 3, we show how we used a Convolutional Neural Network (CNN) for genre classification based on these Mel-spectrograms and use their embeddings for relevant representation on the raw audio input. In further section 4, we will present how we additionally took advantage of further metadata and a final Autoencoder architecture for a final mixed representation for all songs in the database.

The final model will be able to create an embedding vector to represent a given song’s audio and metadata information. Then, it will search in the database for similar songs based on the cosine similarity value of their representations.

## 2 Architecture pipeline

First of all, we need to specify what we consider as similarity in songs. Similarity metrics do not only exist in the audio input features, but in different characteristics of each song, like its genre or the instruments and emotion it gives may also be considered as similar to some and might even depend on metrics we have not tagged. The quantization of similarity is after a base point subjective. In our project, we will consider it as a factor of metadata features and music genre, combining

The idea behind our approach instead of using single CNN or RNN embeddings for representing each audio input, is mainly due to the ability of the autoencoder to perform effective dimensionality reduction by capturing the essential features in a lower-dimensional latent space. This is beneficial for reducing the complexity of data and speeding up subsequent processing tasks. Another benefit is that we can combine features from different modalities and learn robust representations that are less sensitive to noise and variations in the data, since it uses unsupervised learning. In our project, we will use both methods sequentially, obtaining a final general robust representation from different supervised tasks.

As described, we will try to gather as much relevant information as possible regarding song similarity. Our architecture relies on combining multiple such representations and creating a final merged database that translated all underlying dependencies. Our general pipeline can be seen in Figure 0:

Εικόνα που περιέχει κείμενο, διάγραμμα, σκίτσο/σχέδιο, ζωγραφιά

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 0: Song representation retrieval pipeline

In the following chapters, we will discuss further on the nature if the datasets used, the underlying models and different representations to form our final latent space database.

### 2.1 Datasets

As a first step, we needed a dataset with audio data, so that we could extract the mel-spectograms. We choose to use the mel-spectograms of each audio file, because it presents a discernible way to represent a song’s time and frequency domain features. We will use these mel-spectograms for the CNN genre classification mentioned in the following section. For this reason, we concluded to using the GTZAN dataset from kaggle [1], which contains raw audio input of 10 genres with 100 audio clips each, so we have balanced dataset. Each song is of 30 seconds duration and is divided per genre in different folders.

To extract the mel-spectograms, we utilized the librosa library, which sampled the whole duration of the song at a sampling rate of 22050.

We also used librosa for audio feature extraction to extract the chroma STFT features, MFCC, RMS and the zero-cross rate. We calculated for each output feature its mean and standard deviation and saved the audio features on a separate dataframe.

To get better correlation of the final embeddings, we also needed some extra metadata values. For this reason, we used the Million song dataset found in Kaggle [2]. Further details are explained in relevant section.

### 2.2 Classifier models

As mentioned, we will use a CNN model to classify songs into different genre labels based on the song’s mel-spectogram. Normally used for computer vision tasks, the CNN is also able to extract useful features from a given spectrogram.

Εικόνα που περιέχει στιγμιότυπο οθόνης, πολυχρωμία, γραμμή, γραφικά

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 1: Implementation of CNNs for classification on mel-spectograms. Source [3].

We believe that similar songs also might belong to same genre, so by training a genre classifier we will successfully get representations of songs based on genre. The closest a representation is to another, the more likely it is to belong to the same genre and be similar in general.

The prediction happens with a last layer Softmax function that computes the final probabilities of each class. The GTZAN dataset used to train the model contains 10 genres which we use as target labels:

*disco, metal, reggae, blues, rock, classical, jazz, hiphop, country, pop*

Since our dataset is already balanced with 100 songs per class, we use Stratified split of 80% to create the training, test and validation dataset with 800, 100 and 100 samples respectively.

#### 2.2.1 Representation output

We train our CNN with the train dataset in batches. After training, the model is creating good representations of genre dependencies in regards to input mel-spectograms, so we can use these as representations for similarity between genres.

Our training dataset input dimensions are 64x1024 (number of mel bins x feature length) and the final embedding layer of the CNN after training gives latent embedding of dimensions 256. Essentially, the last Forward layer creates a high-dimensional genre-space that can be used with a similarity metric to find the closest songs. However, genre is not the only factor in song similarity, so we will use these embeddings in addition to more feature representations. In later sections we will explain these additional features and finally how they are all combined for a final representation of our database.

#### 2.2.2 Evaluation

We trained our CNN for 100 epochs for the whole training dataset. We used a best model approach, were we saved the final model to be the one that had the best performance in the 100 epochs.

Εικόνα που περιέχει κείμενο, γραμμή, γράφημα, διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 2: CNN Genre classifier training and validation loss, 100 epochs

As we can see from the loss graph in Figure 2, the best model is obtained at approximately epoch 30, before the validation loss gets out of hand. The training loss decreases steadily and consistently over the epochs and eventually flattens out, which suggests the model is fitting the training data well. We can also see this in the achieved accuracies on the test dataset from the given confusion matrix:

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, τετράγωνο, ορθογώνιο παραλληλόγραμμο

Περιγραφή που δημιουργήθηκε αυτόματα

*Figure 3: CNN genre classification confusion matrix.*

From Figure 3, it is evident that the model achieves great classification ability for most categories, while it mostly confuses classical and rock music label. This could be because of similarity in the spectrograms of the two classes. However, it is an all in all good classifier and we will further proceed with using its embeddings to receive meaningful representations of each song.

### 2.3 Metadata

To embed further information about the songs, we will try to also use metadata information based on the Million Song Dataset [2]. This dataset contains around 50000 songs , 1500 of which contain audio, with 21 different metadata. We filtered down the values to the ones we believe have a relation to song similarity to:

*Track\_id, artist, year, danceability, energy, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time\_signature.*

No further separate encoding is performed on the metadata. Instead, both the CNN embeddings and metadata are used as input to the Autoencoder network, and we obtain the final encoded representations there.

### 2.4 Autoencoder

To create a combination of representations based on the different feature embeddings, we can create an Autoencoder architecture. In this way, the hidden dependencies among all features can be learned and we can map the high-dimension initial input features to a latent space. The input data are the concatenation of CNN embeddings created by the trained classifier and the metadata of each of the 1500 entries, so final data dimensions of Nx270 are inputted to the network.

Our Autoencoder consists of the encoder and decoder layer. Each consists of 3 Linear layers of output dimensions 128, 64, 32 and 64, 128 and 270 respectively. The model will be trained to output a decoding as close to the initial input as possible. For training, we used a standard MSE loss with Adam optimizer.

After training we will use the output of its final encoder layer as the final combined feature representation vector, mapping the initial 270 features of each song to a latent vector of shape 32. Each entry is then normalized with a standard scaler for noise pro

## 3. Similarity

After transforming our song database to a new latent representation, we can now infer our model to recommend songs. To do this, we can use distance metric methods such as cosine similarity and Euclidean distance. For this projects, we took advantage of cosine similarity from the sklearn library. Cosine similarity computes the similarity between two songs represented by an embedding arrays X and Y as the normalized dot product of X and Y:

K(X, Y) = <X, Y> / (||X||\*||Y||)

To get the recommendation on a requested song, the database is queried to find its embedding representation based on is name. We then perform cosine similarity to find the 5 songs that are closer in regards to embeddings distance from the whole dataset and return their name to the log output. The higher the similarity distance, the more in common features the songs have.

For evaluating the final recommendation accuracy, we will use manual evaluation for each requested song, since quantifying the accuracy for this issue poses a complex problem and might require training a different Deep Learning model. The experiments of inference are displayed in the below section.

## 4. Experiments

For each experiment, we try to manually evaluate resemblance for 20 different cases. We will request the representation of 20 random different songs from the dataset and ask for the top 5 similar songs. The model’s performance will be manually assessed on whether the recommended 5 songs are close to the requested song’s genre and whether it has any similar metadata info. This will showcase the model’s ability to capture genre similarity, which means it is able to give good low-dimensional representations.

In this section in order not to repeat too much, we will only present you with 3 of the 20 experiments. The rest of the results can be found in the “Top\_10\_songs.csv” in our github:

#### Experiment 1:

**Input query song metadata:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| song name | artist | genre | year | Danc. | energy | loud. | speech. | acoust. | instr. | liv. | Val. | tempo |
| Ndima Ndapedza | Oliver Mtukudzi | World | 1998 | 0.825 | 0.681 | -11.694 | 0.0871 | 0.706 | 0.165 | 0.0988 | 0.954 | 118.678 |

From executing the query, the recommended 5 songs closer to this are:

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά, αριθμός

Περιγραφή που δημιουργήθηκε αυτόματα

Taking a look on the closest recommended song metadata:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | artist | genre | year | Danc. | energy | Loud. | Speech. | Acoust. | Instrum. | Live. | Val. | tempo |
| Sone Otro Mundo | Manu Chao | Latin | 2007 | 0.544 | 0.442 | -12.177 | 0.0489 | 0.0139 | 0.597 | 0.137 | 0.945 | 170.371 |

We can see that the song is of Latin genre, which is very similar to World music. Also, the suggested audio aligns in a close range to the rest of the metadata with the queried song, which means that it is more possible for the songs to align in acoustic features. For example, the loudness and tempo are pretty close, while the valence metric, which also affects the emotion in a track, is almost the exact same.

Taking a look on the rest of the songs features:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | artist | genre | year | Danc. | energy | Loud. | Speech. | Acoust. | Instrum. | Live. | Val. | tempo |
| The Earth Dies Screaming | UB40 | Reggae | 2014 | 0.799 | 0.256 | -14.81 | 0.0686 | 0.0175 | 0.0459 | 0.0753 | 0.418 | 120.506 |
| Homely Girl | UB40 | RnB | 2012 | 0.787 | 0.476 | -11.141 | 0.0575 | 0.194 | 8.00E-05 | 0.0534 | 0.915 | 82.914 |
| Chitlins Con Carne | Kenny Burrell | Blues | 2012 | 0.789 | 0.312 | -14.883 | 0.0436 | 0.676 | 0.225 | 0.115 | 0.846 | 132.764 |
| La VΓ©nus du mΓ©lo | Stacey Kent | Jazz | 2010 | 0.791 | 0.344 | -12.706 | 0.0468 | 0.761 | 3.41E-06 | 0.12 | 0.472 | 124.257 |

Again, the next 2 songs are of genre reggae, which again is of a subgenre of world music. Also notice that the songs are made by the same artist, which means the Autoencoder gives weight on the metadata. All songs are also of the same year period while they are both similar in regards to other characteristics not only to the query but to themselves. In particular, the loudness, danceability, energy are in the same narrow value range.

#### Experiment 2:

Input query song:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | artist | genre | year | Danc. | energy | Loud. | Speech. | Acoust. | Instrum. | Live. | Val. | tempo |
| Remnants | Disturbed | Punk | 2010 | 0.167 | 0.399 | -7.798 | 0.0303 | 0.012 | 0.873 | 0.264 | 0.0388 | 100.059 |

Recommended songs:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | artist | genre | year | Danc. | energy | Loud. | Speech. | Acoust. | Instrum. | Live. | Val. | tempo |
| Idoless | The Distillers | Punk | 2000 | 0.163 | 0.995 | -2.99 | 0.159 | 0.0114 | 0.00271 | 0.187 | 0.147 | 176.891 |
| Waracle | Ayreon | Metal | 2009 | 0.204 | 0.599 | -10.509 | 0.0507 | 0.00313 | 0.0124 | 0.365 | 0.0664 | 121.959 |
| Grease Paint And Monkey Brains | White Zombie | Metal | 2008 | 0.447 | 0.926 | -6.691 | 0.0916 | 6.38E-06 | 0.673 | 0.149 | 0.536 | 146.759 |
| The Gathering | Amorphis | Rap | 1992 | 0.158 | 0.954 | -5.228 | 0.086 | 1.58E-05 | 0.861 | 0.0714 | 0.08 | 65.25 |
| Hospitality | Funeral for a Friend | New Age | 2005 | 0.239 | 0.94 | -4.649 | 0.05 | 3.75E-05 | 0.000779 | 0.142 | 0.615 | 143.98 |

Again, the strongest characteristic of genre is greatly considered by the representations, as similar genres are selected. However the range of the other metadata features is wider, for example the energy metric is a bit far from the query but still in a close distance from one another. Danceability and tempo are the most prominent and similar metrics in this example.

After also listening to the songs we can confirm that the top recommended result is not as accurate, because of false annotation in the initial query song. The query song is not of Punk genre, but rather ballad metal. For this reason, the most recommended song is Punk and more aggressive style than the original query.

Nonetheless, because the model uses merged representation of not only the genre but other metadata, there is still high resemblance in the key and tone to the suggested songs. This can be noticed by the fact that the rest of the recommended songs that focus on matching the metadata are of genre metal with resemblance to ballads, just like the original query.

#### Experiment 3:

**Input query song:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | artist | genre | year | Danc. | energy | Loud. | Speech. | Acoust. | Instrum. | Live. | Val. | tempo |
| Calling Dr. Love | Kiss | Country | 2004 | 0.545 | 0.873 | -8.37 | 0.128 | 0.51 | 2.41E-05 | 0.334 | 0.551 | 127.157 |

Again, the annotation for genre is falsely tagged as Country music, while after inspection it is Rock. So we will look if similar style songs are recommended.

Recommended songs:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| name | artist | genre | year | Danc. | energy | Loud. | Speech. | Acoust. | Instrum. | Live. | Val. | tempo |
| Stay Tonight | Matchbook Romance | RnB | 2004 | 0.399 | 0.958 | -3.778 | 0.0516 | 4.11E-06 | 0.294 | 0.202 | 0.362 | 90.144 |
| Eureka | Arch Enemy | Metal | 1996 | 0.217 | 0.993 | -5.428 | 0.179 | 5.54E-06 | 0.897 | 0.164 | 0.0621 | 120.239 |
| Grease Paint And Monkey Brains | White Zombie | Metal | 2008 | 0.447 | 0.926 | -6.691 | 0.0916 | 6.38E-06 | 0.673 | 0.149 | 0.536 | 146.759 |
| Aqua Dementia | Mastodon | Metal | 2004 | 0.175 | 0.973 | -6.287 | 0.178 | 6.94E-06 | 0.893 | 0.32 | 0.188 | 167.907 |
| Waracle | Ayreon | Metal | 2009 | 0.204 | 0.599 | -10.509 | 0.0507 | 0.00313 | 0.0124 | 0.365 | 0.0664 | 121.959 |

The initial query song is punchy and similar sounding to rock and roll, with blues undertones due to its chord progression.

The most recommended song again is not so much RnB after inspection but rather pop-rock. It is again very punchy and thus the similar energy metric.

This experiment shows more balanced results in all metrics, while all genres are closely related to the original query (punchy rock resembles metal a lot).

## 5. Conclusion

The songs recommended are most of the times belong to the same genre. This is expected as we trained the model based on the notion that two songs are similar to each other if they belong to the same latent representation space that consists one genre class.

While single CNN embeddings are highly effective for tasks like image classification and object detection, autoencoders offer a complementary approach for learning robust, unsupervised representations that can generalize well across different tasks and datasets. This is seen from our experiments, where it is also evident that the use of an Autoencoder to compress information is effective and useful for high-dimension input. Our approach also manages to give accurate results while maintaining low computational cost, something that using single supervised models could not achieve.

## 6. Future Work and optimization

From our implementation, we saw both benefits of supervised embeddings and unsupervised learning. It is evident that the model could benefit on further usage of unsupervised mechanisms to catch unseen feature dependencies that lead to similarity. To further widen this area of no supervision, we could search for ways to find metrics that are not yet in our input dataset.

One way we can learned metric spaces is by using Siamese networks. With a Siamese network, the model is trained to learn a custom metric space that better captures the similarities between songs, given pairs of similar and dissimilar examples. It uses a similar pair of networks with shared weights and map input data to a latent space, where similar input data are mapped close to each other and dissimilar further away. To create similar embeddings for the close data, it uses contrastive loss function.

An easier method that relies on the way we separate the data before applying cosine similarity, is to use a clustering method in the latent representation space to get close embeddings in the same cluster. This will help in more efficient searching, since the search only occurs on the created clusters.

## 7. References

[1] GTZAN Music Genre Classification Dataset, <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>

[2] Million song dataset + spotify API + last.fm dataset, <https://www.kaggle.com/datasets/undefinenull/million-song-dataset-spotify-lastfm/data>

[3] “How I taught a neural network to understand similarities in music audio” ,

<https://medium.com/@silvercloud438/how-i-taught-a-neural-network-to-understand-similarities-in-music-audio-d4fca54c1aed>

[4] Choi, K., Fazekas, G., Cho, K., & Sandler, M.B. (2017). A Tutorial on Deep Learning for Music Information Retrieval, <https://www.semanticscholar.org/reader/6d2b42413839130060cec5c4505a3cb2d15abec7>