Music exploration using machine learning

Jibin Rajan Varghese
Texas A&M University
400 Bizzell St, College Station, TX 77843
jibin@tamu.edu

Mohammed Habibullah Baig
Texas A&M University
400 Bizzell St, College Station, TX 77843
habib_baig.tamu@tamu.edu

Abstract

Music is complex and varies widely in beats, frequencies and structure. This experiment uses the spectrogram of common songs to organize thousands of everyday tracks. We implement various neural network models to train classifiers to segregate neighboring songs with similar features from the mel-spectrogram of the songs. The softmax output of the neural network thus trained is used as a reduced feature set in order to perform dimensionality reduction of the songs. We then create different 2D embeddings of the reduced feature set using various dimensionality reduction techniques. These embeddings are used to place similar song features closer together on a visually interactive songmap. This, in our research so far, is the first attempt at producing a visual and interactive map of music at a full song collection scale that users can use in order to explore the world of music, discover and listen to songs of their preference.

1. Introduction

The relative ease of distribution of music in digital form has lead to the generation of large databases of music in the past few years. Nowadays, listening to music has become very convenient and cheaper than before, hence new music platforms and websites are emerging rapidly, for example, Spotify, Pandora, Douban FM and so on. The Next Big Sound Industry Report 2015[2] indicates that 1,032,225,905,640 (1 trillion) tracks were played in 2015, which, considering each song to be 5 Mb, the overall data size would be 5,161 petabytes. This also implies that the total lifespan of an average person is insufficient to listen and analyze all the songs in the world.Thus many of the major companies in the music industry are starting to use machine learning approaches to classify music, and hence research on music classification and recommendation is catching the attention of a lot of researches in the field.

With people being absorbed in music every day, a visual way to interact and recommend suitable music would greatly enhance user experience with users getting an idea where to look for and discover similar songs of their liking. There are two parts to this problem: recommendation and visualization. Firstly, to recommend suitable songs efficiently, music genres must be identified/predicted with good accuracy. Genres can be thought of as descriptive keywords that convey high-level information about music (Pop, rock, Country, etc). Since songs are too large to be encoded bitwise into a large feature set and processed directly for dimensionality reduction, we need a way to reduce the size of the feature set into a manageable and yet distinguishing subset, which we can generate using neural network and/or deep learning approaches.

We then perform dimensionality reduction techniques on the reduced feature vectors in order to visualize the multidimensional song features into 2D or 3D map of songs. Then the user can explore this map in order to discover similar songs of their taste in the neighborhood of songs they already like, or choose to explore songs in a previously unexplored area in order to discover other types of music. In order to generate visually appealing, yet informative maps, we explore different dimensionality reduction techniques such as Principal Component Analysis[36], T-distributed Stochastic Neighbor Analysis[24] and Uniform Manifold Approximation and Projection (UMAP) [26] with various number of iterations and perplexities.

2. Literature Review

There are two main approaches towards classification of music. The first approach uses music metadata, such as genre, tags, comments, artist information etc to develop a machine learning model to predict missing information or features of a new song, given other related meta-data. For example, if a band "Iron Maiden" is a heavy metal band, songs released by artists in the band would also be in the metal genre.Some of these approaches outlined in [31] include Collaborative Filtering [18], Content based musical information retrieval [5] , using Social Tags [22], Emotion Based Model[20] among many others. These approaches use meta-data appended to the music file in addition to us-

age statistics from users as a form of social tagging.

For genre prediction and recommendation, other systems currently in use are [9] which uses the Friend of a Friend (FOAF) and Rich Site Summary (RSS) vocabularies for recommending music to a user, depending on his musical tastes. One other way would be to have a context-aware recommendation system. Paper [27] exploits the fuzzy system, Bayesian networks and the utility theory in order to recommend appropriate music with respect to the context. These and some other techniques that use lyric based song sentiment analysis [37], [30] have also been shown to perform accurately in classifying music and extracting the theme of the song.

The second approach uses machine learning based approaches to extract sound features from the songs and uses this information to predict features such as timbral texture,beats, rythmic content and pitch content. Some of the recent works in this domain include Multi-Resolution Spectrograms used to extract suitable features from the audio signal on different time scales [13]. A different approach is taken by [23], which uses three feature sets for representing timbral texture, rhythmic content and pitch content respectively, to predict the genre. These features reflected very high information about the music genre since the authors verified it by training statistical pattern recognition classifiers using real-world audio collections. The other approach in [34] uses a latent factor model for recommendation, and predicts the latent factors from music audio using deep neural networks and evaluates the predictions on the Million Song Dataset[7]. It was observed that the predicted latent factors gives sensible recommendations, despite the fact that there is a large semantic gap between the characteristics of a song that affect user preference and the corresponding audio signal. Additionally, Daubechies Wavelet Coefficient Histograms (DWCHs) [23] can also be used, which captures the local and global information of music signals simultaneously by computing wavelet coefficient histograms.

Tools for music exploration have been limited to traditional file system based approaches or play-list approaches. Despite there being various visualizations performed based on self similarity [17] and labeled meta-data [32], these approaches are inherently not scalable to huge volumes of digital music since often annotated meta-data is not available. The visualizations also do not capture the transitive nature of music and apply hard class labels on the types of music. A default classification of music is therefore not accurate, but current state-of-the-art systems do not incorporate this property of transitiveness into music classification, recommendations or visualization techniques, and generate a flat list of similar songs. The discussion regarding how to classify music genres is also ambiguous. Firstly, there is no one-

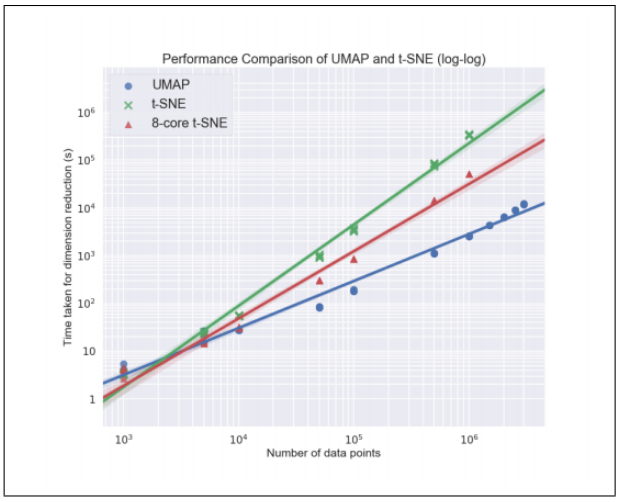


Figure 1: Performance comparison of UMAP vs t-SNE [26]

size-fits-all metric for genre classification as a song might be comprised of intonations of different genres. A common approach used to classify this category of songs, for example in the FMA dataset [12] is to label songs as experimental, or fusion music. While this approach works with traditional genre classifiers without significant loss of prediction accuracy,this approach is subtly flawed because the underlying information regarding the components that comprise the 'Fusion' music is lost. An example would be the case where a song that is a mix of Country and Jazz would be labeled as Fusion, which would be different from a mix of Rock and Metal song which is labeled as Fusion. Also a song could be in between rock music and metal music and could be labeled as metal by one listener , whereas another listener might label the same as rock music. Though both songs are different in the way they sound, common datasets that label music into genres often overlook this fact. Thus in order to capture this transitional information, a new scheme of visualization is needed.

There are various existing approaches to visualize higher dimensional data into two or three dimensions. Though some research from Google and Firenze [16] and have applied mappings of sounds from various sources onto 2D visualizations using dimensionality reduction techniques, these approaches have not yet been implemented on songs, because the size of the feature vectors for a full conversion of a song would be too large for most dimensionality reduction techniques to process. Therefore research like the Infinite Drum Machine [3] and Music-Mappr [21] either visualize very short sounds (1 3 sec) or split a single song into similar comprising sound-clips respectively. These approaches use the t-SNE visualization technique described in [24].It is a variation of SNE that is much eas-

ier to optimize, and produces significantly better visualizations by reducing the tendency to crowd points together in the center of the map. This technique very efficiently visualizes non-parametric high-dimensional data [35] . Another approach to visualize large multi-dimensional data is the Uniform Manifold Approximation and Projection algorithm [26], which uses a framework based on Riemann geometry[15] and algebraic topology. This approach is also scalable because it does not place a computational restriction on the dimensions of the embedding data. A summary of some of the other most popular dimensionality reduction techniques are given in [8].

3. Methodology

In order to achieve the goal of scalability classifying thousands of songs while not loosing its transitive nature, our approach does not perform a one-hot classification on the data. We also develop a means to reduce the size of the musical features from the raw audio of the song using mel-spectogram. We then train a neural network classifier on the reduced data and use the softmax output in order to perform the visualization. The methodology we used is detailed below.

3.1. Audio Feature Extraction using Mel-Spectogram

This process converts the song file into a 1366 x 966 dimension vector. In the pre-processing step, Mel-Spectrograms are generated in a similar manner to [11] whereby we use 96 mel-bins with 256 frame hops at sampling rate of 12000Hz, with the Fast Fourier Transform window size of 512 frames around each frame. For the purposes of this project, we use 30 second clips of songs in order to reduce computation time. Since the exact multiple of audio frames given the above hops,windows and sampling rates for 30 seconds, comes out to be exactly 29.12 seconds, we trim 0.44 seconds of audio from the first and the last portion of the clip. We use the audio processing tool developed by McFee et al [25] in order to perform the required manipulation on the song data. The output of the pre-processing step is a song vector of dimension 1366 x 96 x (total number of songs) which we feed into the neural network model in the subsequent section.

3.2. Song Feature Extraction and Classification

The first step towards generating the map of songs is to extract meaningful features out of them. These would be features like song genre, beats, timbre, language, artists etc. The raw data of the song and its meta-data would be fed into a neural network and trained using classification labels available. In order to do this we used convolutional recurrent neural networl to classify the songs into genres. We then looked for interesting patterns in the outputs of the

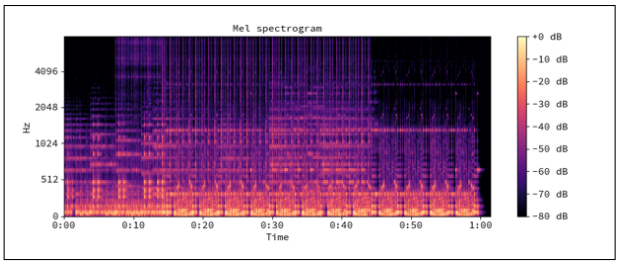


Figure 2: Mel-Spectogram of a song

softmax layer. This approach has already been tried out in neuroscience and vision [29] and we believe using the flattened features of the softmax/penultimate layers of neural networks in music will also provide us with a reduced feature set that can be fed into our dimensionality reduction later on. In order to test this approach, we generate a t-SNE plot directly of the Mel-spectogram of songs. Figure [3] shows that the raw Mel-spectogram does not provide us with a meaningful clustering of songs.

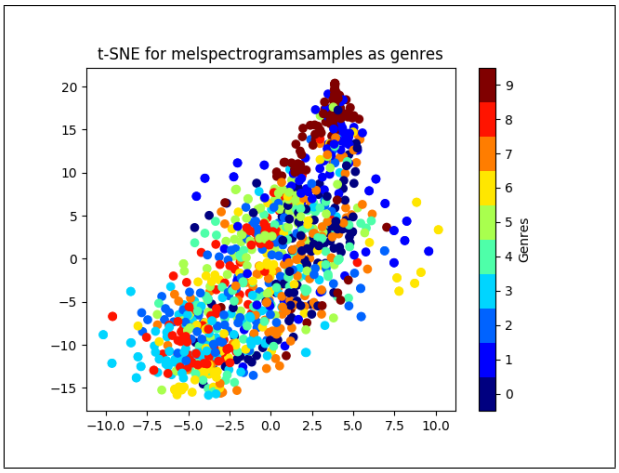


Figure 3: Performing T-SNE of the mel-spectograms of the songs directly does not lead to distinct or distinguishable results

Thus, in order to achieve meaningful dimensionality reduction, the song features are decomposed into a 10 dimensional vector using a convolutional recurrent neural network. We experimented with the size and depth of the neural network in order to develop a suitable classifier for genre prediction, with a targeted accuracy of atleast 0.5. We tried models from [11] and [**?**] and finally built our own model. Using the right features are very important to identify genres [38] or in our case the right feature vectors of the songs. Here we need to balance out the specificity, ie. response to particular type of song and the generality, ie. the amount

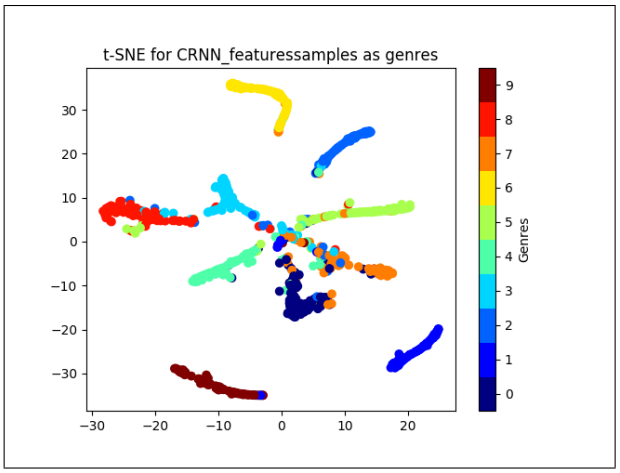


Figure 4: Performing T-SNE of the mel-spectograms of the songs using the softmax output results in better classification and clustering

of spread of feature vectors when the songs are fed into the system, so that we get a decent but relevant spread in our final song-map. If the outputs of the song classifier are too specific, the final dimensionality reduction will have a lot of data points clustered close together and there will not be sufficient scatter in the data-points in the visualization. On the other hand data that is poorly classified will have a lot of intermixed points which will lead to poor relevance of one song to other. Thus the aim of the project is not to exceed the performance metric of the existing classification systems, but to tune the results of the neural networks to generate an appropriate set of feature vectors, which upon being fed into the dimensionality reduction system provides us with a map of music which will be relevant to the user.

The final model we developed was a 8 layer Gated Recurrent Convolutional Neural Network with 6 convolutional layers and 2 recurrent layers. The convolutional layers extract high level features from the neural network while the recurrent layers provide the ability to capture time-series information from the data. In this manner, features such as beats, treble, and vocals can be thought of as a series of neuron activations in a sequential manner, with the convolutional layers aggregating lower level features into higher level features, and the recurrent units firing, if a particular high-level feature is sustained for a fixed period of time, for example legato, staccato and vibrato in vocals[1]. Paper [6] explains some of these patterns in violin music.

Even after mel-spectogram reduction, the size of the input was very large. Thus we optimized the algorithms in order to reduce the computation time . We converted the tensorflow backend of keras to GPU based code in order to reduce computation time from approximately 5 minutes for

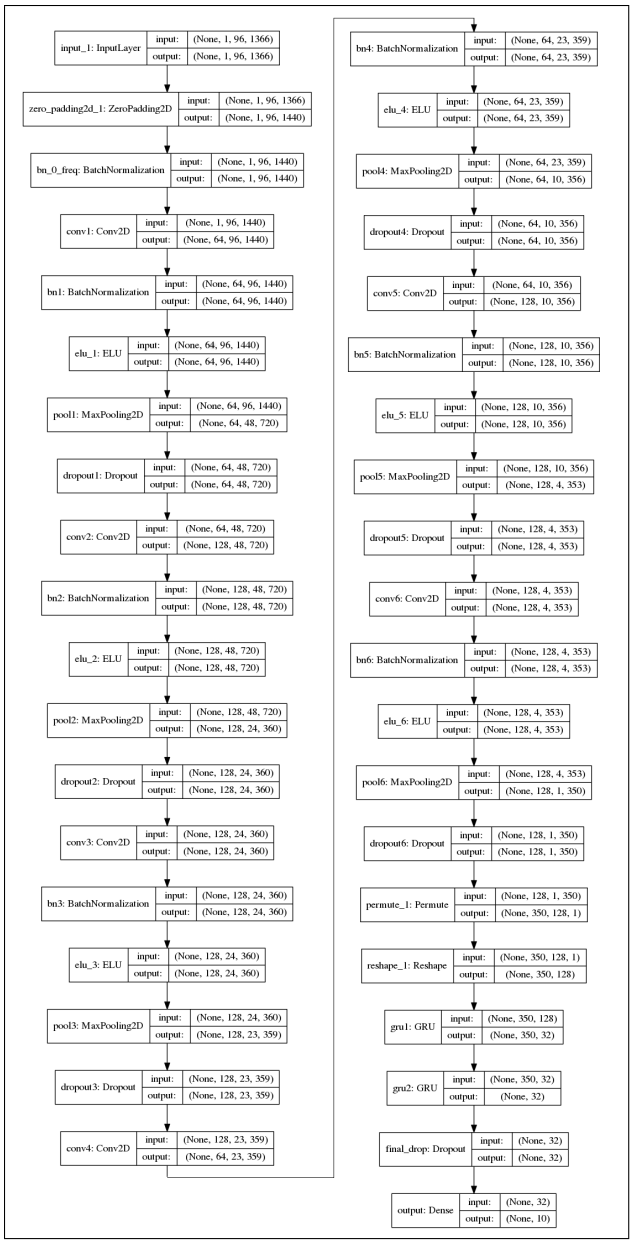


Figure 5: Final CRNN model that gives 80% accuracy on test data

1 epoch of 1000 songs, to around 40 seconds per epoch for 1000 songs. This was done by lowering the batch size and running the computation using an Nvidia GTX1080x GPU. This allowed us to scale up the data collection from 400 to 1000 to upto 23,000 songs.

A table detailing the architecture of various models and their classification accuracy, with respect to various hyper-parameters are given.

Model	Dataset	Epochs	Train Acc.	Vali. Acc.	Test Acc.
4 conv. layers	GTZAN	40	0.87	0.8	0.78
4 conv. layers	GTZAN	300	0.89	0.78	0.75
6 conv. layers	GTZAN	100	0.93	0.65	0.55
6 conv. layers	GTZAN + FMA	100	0.90	0.80	0.80

Table 1: Accuracies for different neural network models

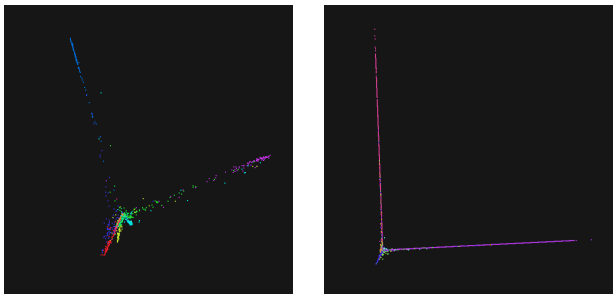
3.3. Visualization

There are multiple methods to perform dimensionality reduction. Principal Component Analysis [36] and Linear Discriminant Analysis [19] are two commonly used methods to reduce multidimensional features into two or three dimensions.PCA provides an intuitive understanding of the music map for a starting point.In our experiment(Fig. 6(a)),the transitions of various genres of music form two main outward branches, one being Metal and the other being Hip-hop. Pop music gradually transitions into hip-hop, while rock music transitions into metal. Classical, country and reggae music appear as three branches forming the base of the Y stem which merges in the center. This trend is still valid on music on 1000 song dataset or the 23,000 song dataset.

However, PCA as shown in Fig 6.b also indicates that the map becomes less intuitive as the dataset size increases and the number of available songs in one or two genres have a larger majority than the other. Thus PCA, though providing an initial understanding of the way music is comprised, does not scale well to large datasets due to its many inherent limitations.For example, PCA is a bad choice because it is a linear algorithm which cannot distinguish non-linear structures in the data. An in-depth study of these limitations is performed by [28], albeit for a different use case. Hence we are exploring the use of t-SNE [24] and UMAP [26] in order to develop the music map.These techniques perform much better than PCA for large songs, as can be seen in the results section(Figure 10).

3.4. User Interface

Finally we use the music map generated by the visualization into an interactive user interface that allows the user to click and play the songs in the area selected. To accomplish this task, we used the popular web browser rendering tool Three.js [14] in order to render the songs. The input to the web application is a json file of coordinates and paths of each song in a nested list dictionary structure. These 'embeddings' are generated for all the songs present in the database, in order to get the best possible visual output. The points encode a GET request to the URL of the song in the database, which is played back using the Web Audio API



(a) PCA on GTZAN dataset of 1000 songs

(b) PCA on GTZAN+FMA dataset of 23,141 songs

Figure 6: PCA plots of 10 different genres of music show that this method is not scalable, although it is easy to comprehend the transition between different genres of music.

[10]. We also used another extension to Web Audio API called webaudiox.js in order to add additional functionality such as audio volume normalization, crossfading and sound smoothing in order to make the songs more appealing to the listener. Note that all songs of a particular genre have the same colour. The web page was designed using a template from [4].

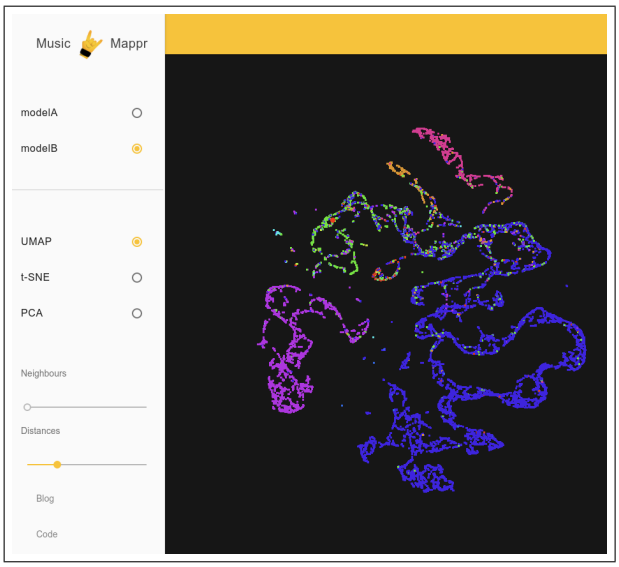


Figure 7: User interface of our application

4. Datasets Used

There are millions of songs in the world to chose from and add to our system. With an impartial feature classifier, we hope to get better and better at mapping songs to the point where any new song released can be added to the system and it will put the song into the right spot on our

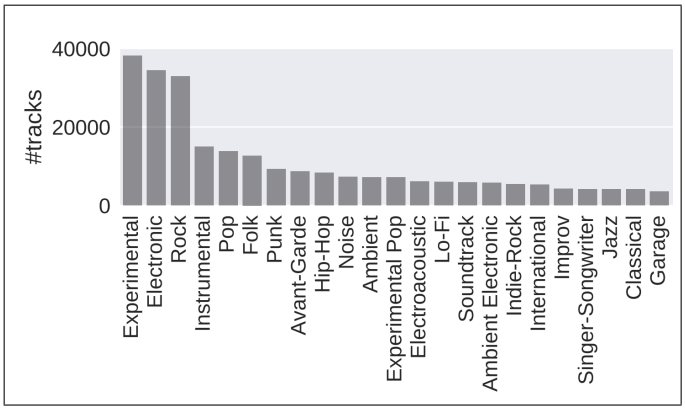


Figure 8: Distribution of no.of tracks per Genre in FMA dataset

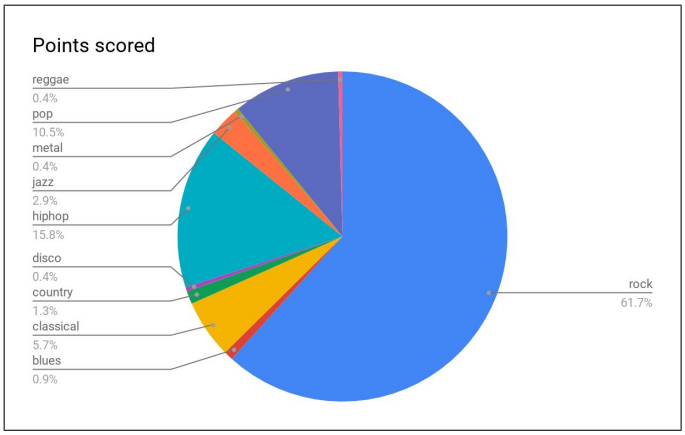


Figure 9: Distribution of no.of tracks per Genre used in our music map after filtering

map. However, given the limitations of time and computing power, we start with smaller song sets and learn the features on smaller subsets of music. Incrementally, we worked on the following datasets :

1. GTZAN Dataset : This dataset is the most-used public dataset for evaluation in machine listening research for music genre recognition (MGR). It was first used for the well known paper in genre classification [33]. The dataset consists of 1000 audio tracks each 30 seconds long distributed among 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .wav format.

dataset	clips	genres	length[s]	size[GiB]
small	8,000	8	30	7.4
medium	25,000	16	30	23
large	106,574	161	30	98
full	106,574	161	278	917

Table 2: FMA data subsets

2. Free Music Archive Dataset: We wanted to train our Neural network model on relatively bigger dataset than GTZAN. Although Million Song Dataset (MSD)[7] as well as the newer AudioSet and AcousticBrainz are very large-scale reference datasets, they do not provide raw mp3 files and downloading the mp3 files in itself is a challenge due to copyright issues. FMA provides an excellent alternative as a medium scale music dataset of more than 100k songs with freely available full-length and high-quality audio. It provides precomputed features, together with track- and user-level meta-data, and tags. The dataset is divided and maintained in subsets listed in Table 2.

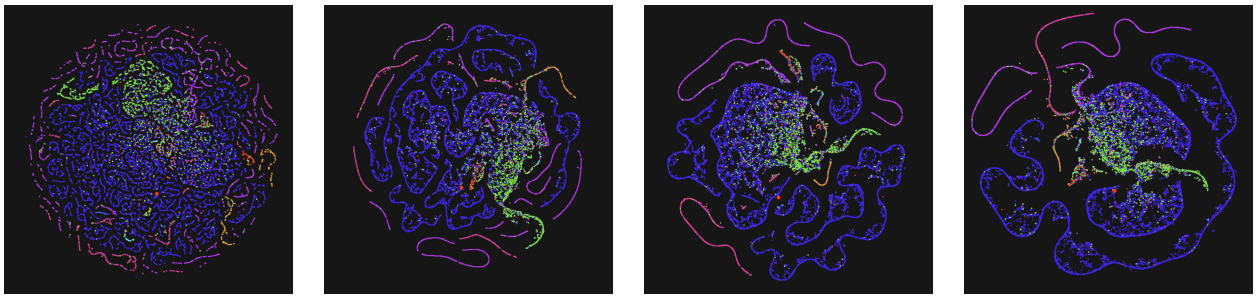
The problem with the FMA dataset was that there was a non-uniform distribution of songs in different genres(Figure 8). Hence, we merged both GTZAN and FMA datasets together and selected 10 popular genres and filtered the songs to be used for our visualization. Distribution of trakcs per genre was still dominated by few genres but it has helped us further our analysis on Genres selected from GTZAN dataset (Figure 9). Finally, the filtered dataset consisted of 23,141 songs which were divided among training testing and validation set, with 14,809 songs for training, 3,703 for validation and 4,629 for testing.

5. Results

Plotting songs on a 2D map gave us many auditory insights which are hard to describe on paper and vary from person to person. On the other hand, visual properties are easier to describe in general. For example, the t-SNE algorithm has a hyper-parameter - Perplexity, which balances attention between local and global structures of the data. It specifies the number of close neighbors each point has. We observed that as we increase the perplexity value we started getting more visually distinct clusters of music genres.

UMAP has two hyper-parameters: no-of-neighbors, which indicates the number of neighboring points in the local approximations of the manifold structure. If more neighbors are considered, lower dimension preserves global structure in dataset. The second parameter is min-dist, which controls how tightly the embedding is allowed to compress points together. If compressing is allowed with higher margin it ensures that the embedded points are more evenly distributed and retain the global structure in dataset.

It was observed that with larger values of both no-of-neighbors and min-dist we started getting much better clusters of music genres on our dataset. In conclusion, we found

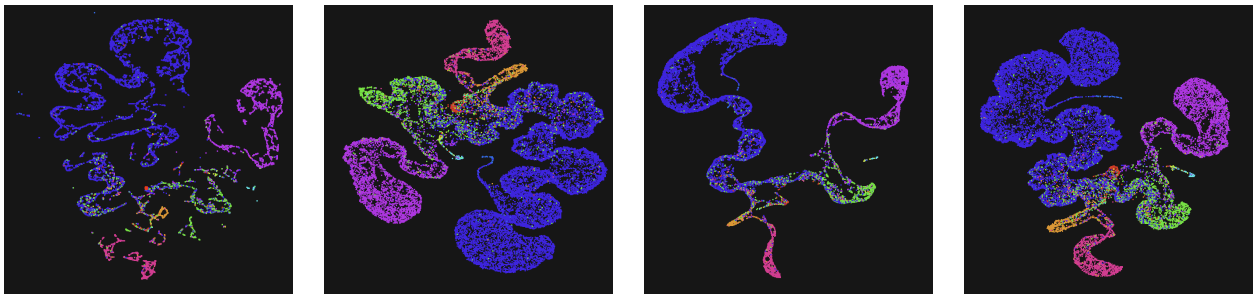


(a) t-SNE perplexity=5, Iterations=500

(b) t-SNE perplexity=30, Iterations=500

(c) t-SNE perplexity=50, Iterations=500

(d) t-SNE perplexity=100, Iterations=500



(e) UMAP neighbor=5, min-dist=0.01

(f) UMAP neighbor=30, min-dist=0.5

(g) UMAP neighbor=50, min-dist=0.1

(h) UMAP neighbor=50, min-dist=0.5

Figure 10: Music Maps generated using different t-SNE and UMAP parameters

that the distribution of points by UMAP is more visually appealing than t-SNE to most viewers. Figure 10 shows the various outputs generated by the system for varying parameter combinations.

6. Conclusion

We present a novel method of visualizing music using the latest state-of-art visualization techniques, powered by a custom built neural network that matches the current state-of-art in classification accuracy, and is optimized to run an epoch of 1000 songs in under a minute. This enables us to develop the first mapping of music on a full-song collection level in current literature, to the best of our knowledge. The above methodology is successfully demonstrated on a dataset of 23,000 audio tracks which demonstrates the viability of the idea that the softmax output of the neural network can be used as a reduced feature set for song visualization. The outputs of the visualization algorithms, are not only mysterious but also beautiful, presenting the audience with an unprecedented way of understanding music, opening up a new dimension in how music is perceived and on a larger scale, interacted with. This, we believe is the greatest outcome of the project.

7. Activity Split-up

Although we did most of the project sitting together in a pair programming style in our lab, this is the rough split up of tasks and responsibilities.

- Conceptualizing the idea - Jibin Rajan , Mohammed Habib
- GTZAN preporcessing - Jibin Rajan
- FMA preprocessing - Mohammed Habib
- Generating Mel-spectogram - Jibin Rajan
- Implementing multiple existing models - Jibin Rajan, Mohammed Habib
- Evaluation and finalzing our own model - Mohammed Habib
- Optimizing code for running on GPU - Jibin Rajan
- Plotting t-SNE from softmax - Mohammed Habib
- Creation of TSNE embedding in json - Jibin Rajan
- Server setup and webaudiox for playing music- Mohammed Habib
- Visualization of json in demo.html - Jibin Rajan
- Code cleanup and refactoring - Mohammed Habib
- Documentation - Jibin Rajan
- Final report - Mohammed Habib, Jibin Rajan

References

[1] 3 voice techniques: Legato, staccato, vibrato: https://www.youtube.com/watch?v=shtz01-zie8.

[2] Data to date: the rapid rise of social and streaming: https://www.nextbigsound.com/industry-report/2015.

[3] Google infinite drum machine: https://experiments.withgoogle.com/ai/drum-machine.

[4] Umap tsne embedding visualiser: https://github.com/fedden/umap-tsne-embedding-visualiser.

[5] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[6] L. Aversano. Terminologia violinistica tra sei e settecento. *Tra le note*, pages 1000–1034, 1996.

[7] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Ismir*, volume 2, page 10, 2011.

[8] K. Bunte, M. Biehl, and B. Hammer. A general framework for dimensionality-reducing data visualization mapping. *Neural Computation*, 24(3):771–804, 2012.

[9] Ò. Celma, M. Ramírez, and P. Herrera. Foafing the music: A music recommendation system based on rss feeds and user preferences. In *in ISMIR*. Citeseer, 2005.

[10] H. Choi and J. Berger. Waax: Web audio api extension. In *NIME*, pages 499–502, 2013.

[11] K. Choi, G. Fazekas, M. Sandler, and K. Cho. Convolutional recurrent neural networks for music classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 2392–2396. IEEE, 2017.

[12] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

[13] S. Dieleman and B. Schrauwen. Multiscale approaches to music audio feature learning. In *14th International Society for Music Information Retrieval Conference (ISMIR-2013)*, pages 116–121. Pontifícia Universidade Católica do Paraná, 2013.

[14] J. Dirksen. *Learning Three. js: the JavaScript 3D library for WebGL*. Packt Publishing Ltd, 2013.

[15] P. Filzmoser, K. Hron, and C. Reimann. Univariate statistical analysis of environmental (compositional) data: problems and possibilities. *Science of the Total Environment*, 407(23):6100–6108, 2009.

[16] S. Flesch, A.-S. Gutsche, and D. Paschen. Exploring the untapped potential of sound maps.

[17] J. Foote. Visualizing music and audio using self-similarity. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 77–80. ACM, 1999.

[18] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201. ACM Press/Addison-Wesley Publishing Co., 1995.

[19] A. J. Izenman. Linear discriminant analysis. In *Modern multivariate statistical techniques*, pages 237–280. Springer, 2013.

[20] F.-F. Kuo, M.-F. Chiang, M.-K. Shan, and S.-Y. Lee. Emotion-based music recommendation by association discovery from film music. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 507–510. ACM, 2005.

[21] S. W. Lee, J. Bang, and G. Essl. Live coding youtube: Organizing streaming media for an audiovisual performance. *Ann Arbor*, 1001:48109–2121, 2017.

[22] M. Levy and M. Sandler. A semantic space for music derived from social tags. *Austrian Compuer Society*, 1:12, 2007.

[23] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 282–289. ACM, 2003.

[24] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[25] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[26] L. McInnes and J. Healy. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[27] H.-S. Park, J.-O. Yoo, and S.-B. Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *International conference on Fuzzy systems and knowledge discovery*, pages 970–979. Springer, 2006.

[28] S. Prasad and L. M. Bruce. Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geoscience and Remote Sensing Letters*, 5(4):625–629, 2008.

[29] S. Rohit and S. Chakravarthy. A convolutional neural network model of the neural responses of inferotemporal cortex to complex visual objects. *BMC neuroscience*, 12(1):P35, 2011.

[30] S. Saxena and C. J. Romanowski. Theme extraction from lyrics.

[31] Y. Song, S. Dixon, and M. Pearce. A survey of music recommendation systems and future perspectives. In *9th International Symposium on Computer Music Modeling and Retrieval*, volume 4, 2012.

[32] M. Torrens, P. Hertzog, and J. L. Arcos. Visualizing and exploring personal music libraries. In *ISMIR*, 2004.

[33] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[34] A. Van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.

[35] M. M. Van Hulle. Self-organizing maps. In *Handbook of Natural Computing*, pages 585–622. Springer, 2012.

[36] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[37] Y. Xia, L. Wang, and K.-F. Wong. Sentiment vector space model for lyric-based song sentiment classification. *International Journal of Computer Processing Of Languages*, 21(04):309–330, 2008.

[38] E. Zheng, M. Moh, and T.-S. Moh. Music genre classification: A n-gram based musicological approach. In *Advance Computing Conference (IACC), 2017 IEEE 7th International*, pages 671–677. IEEE, 2017.