Using Music Self Similarity Matrices to Identify the Genre of Music

Mazin Bokhari
mbokhar2@illinois.edu

Abhishek Modi
akmodi2@illinois.edu

Stephen Sullivan
sksulli2@illinois.edu

University of Illinois at Urbana - Champaign

Abstract

We present a novel method of identifying the genre of a song by extracting important features and structure of music and the learning problems associated with this method. Most current methods of genre identification rely on identifying the song itself and using the tags associated with the song to identify the genre. Using acoustic properties of the song for genre identification is not as common. Our approach uses the acoustic similarity within a song represented as two-dimensional self-similarity matrix by using a quantifiable similarity between every two fixed-length windows of time of each song. We use similarities in such matrices to train classifies and make decisions about the genre of the song.

Keywords

Signal processing, neural networks, machine learning, similarity matrix

Introduction

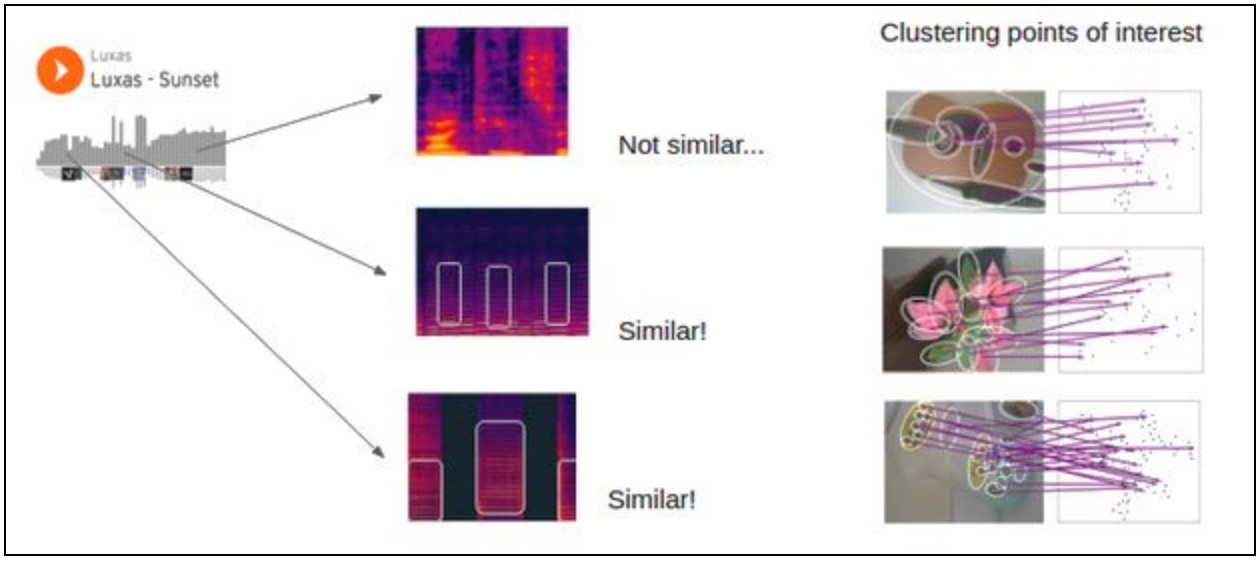
Data science concerning audio is a relatively new field, relative to other types of machine perception, but it has enjoyed rich development recently as the myriad applications for prediction, identification and synthesis of audio have become apparent. Music recommendation systems provide a very lucrative outlet for state of the art data science techniques [6], as does audio isolation [5]. Classification of music, of particular importance to recommendation systems, music identification systems and artificial intelligences [4], requires the capability to make high level qualitative statements about music, such as the genre of a piece, the major structural elements of a piece or a judgement of similarity between two pieces. Our work focuses on enabling the automatic creation of these types of statements by a machine perceiving audio.

Our analysis techniques draw upon previous research in the field of music visualization [3], primarily aimed at intelligent feature extraction from music, upon which established statistical learning techniques can be used. Our approach tests very simple models, support vector machines, touted for their practical performance and ease of use [2] along with much more sophisticated deep learning models employed by state of the art systems attempting to make high level qualitative judgments in other domains, especially in computer vision [7].

We present a novel method of classifying music by genre, using self similarity matrices as a method for feature extraction for songs. As presented in [3], it is clear that the acoustic similarity within a song can be represented by a two-dimensional self-similarity matrix. Our thesis is that the features made apparent by such a matrix can be used to identify which musical genre a song could be classified as. This is based on the fact that music of different genres generally have a recognizably different tempos, different sets of instruments, and different chorus structures, etc. These are are features that can be visually identified when looking at the self-similarity matrix representation of the song.

We compute the self similarity matrices of songs and try various machine learning techniques in order to cluster some of these characteristic structures of genre macroscopically and use this to predict and label the genres of new songs.

Previous Approaches to Genre Identification



Existing common approaches to genre identification require that the song is identified. The song's identity is then used to find the song's associated tags which contain information

about the genre. The obvious drawback of these approaches is that we are unable to identify the genre of new songs.

The song identification is commonly done by creating a number of spectrogram representations of the songs [3]. A clustering algorithm would often be run to find points of interest in the spectrograms. This result is used as a fingerprint for the song, represented as a point cloud in N dimensional space, where N is commonly the number of bins used in the original fourier transform.

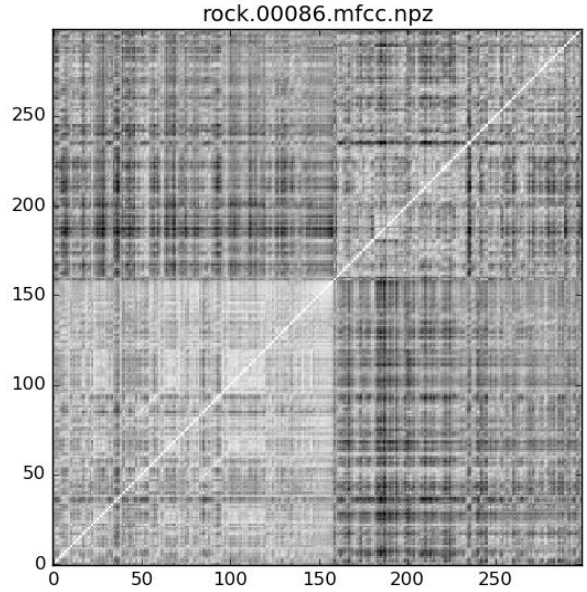
Another set of points are similarly computed for some small contiguous excerpt of the input song (where the input song is the one that is to be identified). Shazam for instance, requires 10 seconds of the input song. The set of points for the input (input fingerprint) are then compared with the precomputed fingerprints of known songs using. Nearest Neighbours algorithms work well for this. Typically an energy optimization function is then used to make a decision for matching the input fingerprint with one of the precomputed fingerprint and thus, identifying the song.

In this approach, structural patterns of the song are not used to make decisions.

Proposed Approach - Using the SSM

Our approach attempts to improve on previous approaches by analysing the structure of the song. This is to say that we do not try to identity the song. Thus, we hope to be able identify the genre of new songs as well as known songs.

We analyse the structure of a songs by generating a Self Similarity Matrix (SSM) [3] for it. These matrices make a number of features of the song visually apparent. These features include repetition, layout of chorus, tempo, etc.



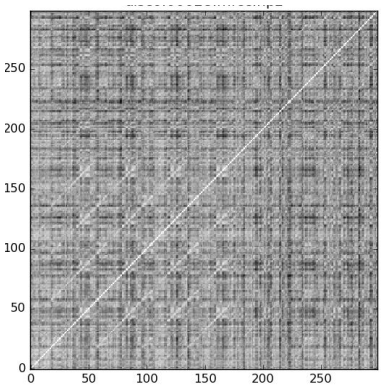
By generating SSMs for a large number of songs we can train classifiers to identify the genre based on the generated SSM for a new song.

Dataset Used

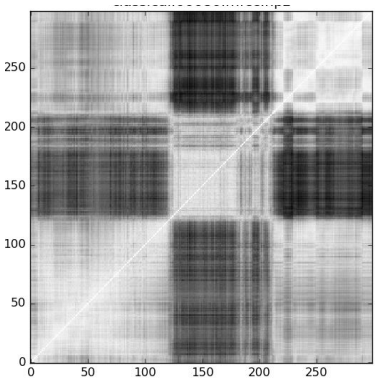
The dataset used is the GTZAN, which is comprised of 1000 labeled songs in the public domain uniformly distributed across 10 different genres. Further insights were gathered by generating our own noiseless audio samples, which provide an idealized form of input for a signal processing and learning task.

Self-Similarity Matrices for Feature Extraction

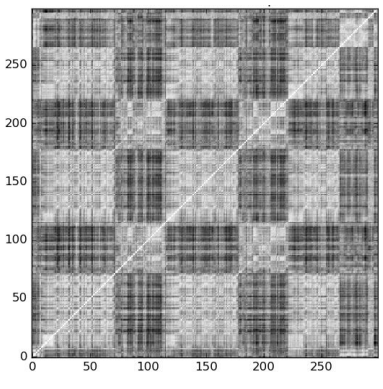
While we first try to reproduce these results with a simple Fourier transform of each song, we end up using the same technique of generating the Mel-frequency cepstral coefficients for each song, and then computing the dot product over a fixed-length window to see the acoustic similarity between any two points in a song. The results are quite convincing, as we can see qualitatively where a song is most similar with itself by looking at the white spots in a visualization of the self-similarity matrix.



Disco



Classical



Blues

From left to right, the self similarity matrix for a disco, classical and blues song. Note the stark contrast in visual structure across genres.

It follows, then, that there is a completely white diagonal within the self-similarity matrix, which is exactly where we would expect a song to be most similar with itself. More interesting parts of the song, such as the chorus, also have clear patches of white, as we would expect. As aforementioned several characteristics about a song, such as consistency of

tempo and types of instruments used across the song, can be hidden quantitatively in the self-similarity matrix.

Machine Learning Techniques Attempted

We now delve into the topic of modeling the features of structural similarity in an attempt to learn to classify based on genre. Two main techniques are attempted in this paper, the support-vector machine and neural nets. In order to measure the accuracy of our results, we introduce the problem set formally: the dataset contains 1000 labeled songs across 10 different genres, although we use random 20/80 test-train splits in order to reduce overfitting.

More generally, we could say that our test set has k samples, and there are d genres. Thus, the expected accuracy of correctly classifying songs randomly can be obtained as follows:

Much like the ball and bin problem, the probability of correctly classifying the genre of song s_i (for $i \in [1,k]$) is $\frac{1}{d}$, as there is a uniform probability of it belonging to any genre (a priori). Call this event X_i. Then, for song s_i, X_i is a Bernoulli random variable with probability $\frac{1}{d}$. The proportion of songs accurately classified is then $P = \frac{\sum_{i=1}^{k} X_i}{k}$. The expectation of which gives $E[P] = E[\frac{\sum_{i=1}^{k} X_i}{k}] = \frac{E[\sum_{i=1}^{k} X_i]}{k} = \frac{\sum_{i=1}^{k} E[X_i]}{k} = \frac{\sum_{i=1}^{k} \frac{1}{d}}{k} = \frac{k\frac{1}{d}}{k} = \frac{1}{d}$. Using $d = 10$ as the number of genres, we see that the expected accuracy by classifying randomly is $\frac{1}{d} = \frac{1}{10} = 0.1 = 10\%$

Our goal to is to beat this figure by a great deal. Below we discuss the various methods we tried for doing so.

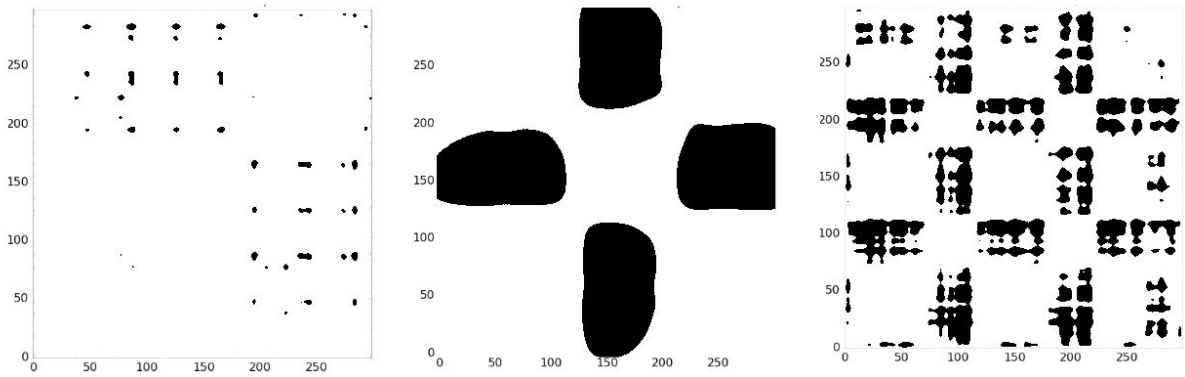
Analysis Zero

The difficulty in classifying songs based on the SSM arises from the fact that the SSM is not a collection of feature values like what might be seen in a more traditional dataset; i.e. it is not obvious how the SSM should be interpreted in order to produce column vectors that can be passed to most traditional machine learning machinery. Our first attempt was to project the SSM to a one dimensional value, the average similarity of a song, determined by averaging together all values in the SSM. We then use a linear SVM to partition clusters of points in one dimension into genres. Our expectation was that this technique may not be better than random guessing.

Patch Matching

For this approach, we reduced the generated SSM into a picture of a Kernel (the kind used in Computer Vision) or blobs.

Upon converting the previous three SSMs into these blobs, we get the following result.

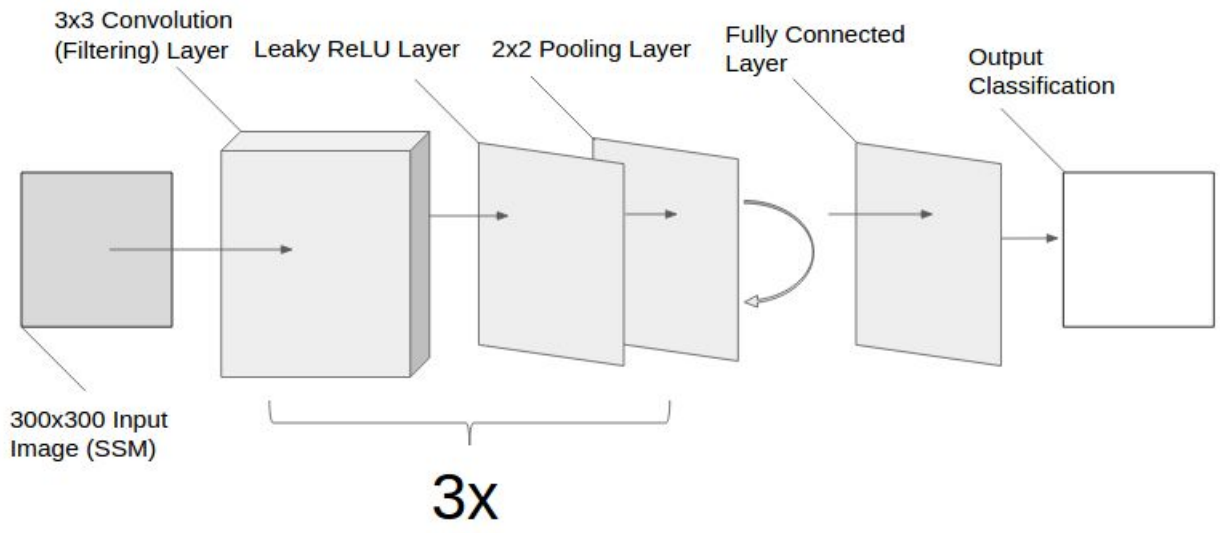


These patches are found to be characteristic of the genres. We then run patch matching algorithms to try to identify these shapes in SSM of the input song.

With more training data, we can also learn some generalized shape or perhaps some set of generalized shapes for each genre. These can then be used to improve the results from the patch matching algorithm. However this refinement is relegated to future work.

Convolutional Neural Nets

Since SSMs have an inherent visual structure to them, and due to the compelling evidence [1] that accurate image classifiers can be built based on Convolutional Neural Networks (CNNs), our final technique involved classifying songs' SSMs using CNNs. Based on learned image filters, CNNs are best applied when extracting successively more complex structures from images [8]. This type of analysis allows our techniques to recognize similarity between SSMs in a position and transformation independent manner. Our simple CNN architecture consists of the following layers:



Note the use of Leaky Rectified Linear Units (ReLUs) as opposed to traditional ReLUs. Our choice reflects recent literature [8] attempting to mitigate the dangers of dead ReLUs in neural nets.

The overall architecture aims to capture three levels of detail from each SSM. The convolutional layers condense down structural information from the input image or a previous layer into more simplified representations, ultimately representing a final classification for the SSM. These classifications represent our aforementioned genres.

Results and Conclusions

The success our analysis techniques is measured by the classification accuracy of each technique, using the class confusion matrix for each classifier. Again, we compare the classification accuracy of each classifier versus the baseline performance of random guessing: 10% accuracy. Our classifiers were evaluated using held out data averaged over several 20/80 test train splits.

Analysis Zero

Successful at performing better than the baseline, but barely so, classifying with 14% accuracy on average. This classifier may be improved by using a multi-dimensional projection of data instead of a single dimensional projection, as this discards a significant amount of data stored in each SSM. For instance, choosing N values for each SSM, where an average value is recorded for N tiles over the SSM, might improve this technique.

Patch Matching

This method did not fare well (8%) either for known songs or for new songs if the input song length was different from the training song length. If the lengths were the same, this method fared better for known songs (35%) but still poorly for new songs (6%). In summary, we see that this method created a classifier that was overfit to the training data.

This is because of the selected patch matching algorithm. This algorithm searched for similar structures in the entire image but did not search for subsets of the shape. This is to say that overlapping repetitions of shapes would be treated as different from a single instance of the shape. A different patch matching algorithm can perhaps be used for better results. This is delegated to future work.

Another refinement to this would be to create a CNN where the convolutional layer creates the blobs and the pool layer essentially runs patch matching. This is also delegated to future work.

Convolutional Neural Net

Much more successful at outperforming the baseline, classifying with 28% accuracy on average. While this performance is ultimately not as accurate as state of the art generalized genre classifiers [10,11] that classify with up to 97% accuracy, the performance is well above the baseline, indicating that CNN analysis of SSMs may be a viable means of determining the genre of general sets of songs. Our CNN ran for between 5 and 2000 training steps, where performance improved between 5 and 100 iterations, but saw negligible gain between 100 and 2000 iterations, suggesting that our simple architecture might be improved with a deeper network. Further research might test different rectifier units, such as traditional ReLUs, randomized ReLUs, or parametric ReLUs, all of which may improve performance, but affect training time and require slightly different training procedures [8].

Possibly aiding all three techniques, the input audio could be passed through filters isolating certain frequency ranges, generating N SSMs for N frequency ranges. We expect that this technique might aid Analysis Zero in becoming a valid analysis of SSMs, as it would help increase the dimensionality of the processed data, decreasing information loss when transforming the SSM into a dataset easily processed by a linear SVM.

Future Work

The CV approach using Kernels and Blobs was promising. However the result was suboptimal because of the choice of patch matching algorithm. A different algorithm can be used which can find subsets of patches / blog structures. Perhaps the current approach can be used in conjunction with a CNN to improve results as well.

In addition, the CNN could be modified with various flavors of ReLUs in the architecture, possibly employing traditional ReLUs, parametric ReLUs, or randomized ReLUs. In addition, a deeper network could be formed, maybe doubling the number of rectify and convolutional layers in each group, or adding more groups. Since the design of convolutional neural nets is an open question in computer science, there may be many possible improvements over our architecture.

Finally, preprocessing the data fed to each classifier could improve our results. Performing PCA or ICA on our data might yield better classification accuracy, as could generating multiple SSMs for bandpassed versions of the source audio. This would lead to higher dimensional projections in case of SVM analysis, and more input data for CNNs.

References

[1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks.” *University of Toronto.* 2012.

[2] Blanchard, Gilles. "Statistical Performance of Support Vector Machines." *The Annals of Statistics* 36.2 (2008): 489-531.

[3] Foote, Jonathan. "Visualizing Music and Audio Using Self-Similarity." (n.d.): n. pag. FX Palo Alto Laboratory, Inc.

[4] Hawkins, J., and Blakeslee, S. *On Intelligence*. N.p.: n.p., 2004.

[5] Huang, Po-Sen, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. "Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation." *IEEE/ACM Transactions on Audio, Speech, and Language Processing IEEE/ACM Trans. Audio Speech Lang. Process.* 23.12 (2015): 2136-147.

[6] Oord, Aaron, Sander Dieleman, and Benjamin Schrauwen. "Wavelet-Based Music Recommendation." *Proceedings of the 8th International Conference on Web Information Systems and Technologies* (2012).

[7] Derek Hoiem. “Object Recognition and Augmented Reality.” *CD 445: Computational Photography, University of Illinois at Urbana-Champaign.* Web.

[8] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going Deeper with Convolutions." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015).

[9] Xu, Bing, Naiyan Wang, Tianqi Chen, and Mu Li. "Empirical Evaluation of Rectified Activations in Convolution Network." (n.d.).

[10] Liang, Dawen, Haijie Gu, and Brendan O'Connor. "Music Genre Classification with the Million Song Dataset." (n.d.): Carnegie Mellon University.

[11] Ghosal, Arijit, Rudrasis Chakraborty, Bibhas Chandra Dhara, and Sanjoy Kumar Saha. "Perceptual Feature-based Song Genre Classification Using RANSAC." IJCISTUDIES International Journal of Computational Intelligence Studies 4.1 (2015): 31.