Principia Musica

Zach Wolpe, Tiffany Woodley

March 2020

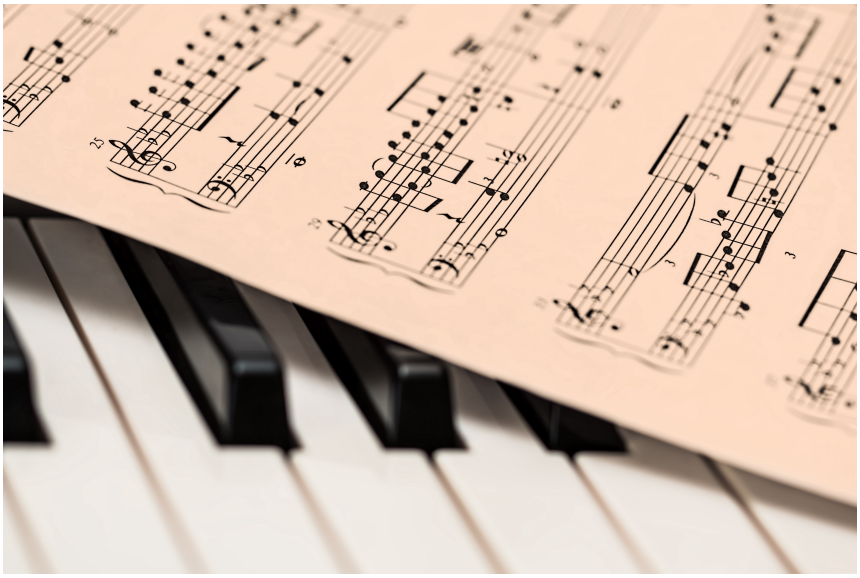


Figure 1: Human readable musical representation

Contents

1 Abstract

Analogies are often drawn between music and mathematics, alluding to the underlying structure of music. It then naturally follows that if we consider music as a high-dimensional numerical representation, we ought to be able to apply statistical methods, perform inference and model music.

The first challenge when conducting statistical techniques on non conventional data types is to adequately and efficiently describe the data numerically without exponential dimensionality expansion.

Any sound can be visualized graphically as a sound wave. Representing an unprocessed sound-wave as a matrix (encapsulated by image pixel density) provides a very sparse and irregular domain, thus correlations are seldom found without various data simplifications and dimensionality reduction techniques - in this paper we describe the most proven and useful techniques to represent music in a useful numerical setting - these techniques are then subsequently used to prepare audio data for statistical computation and analysis.

Sufficient simplification and representation of the data allows one to then explore various clustering and modeling techniques. Thus adequate description of audio allows one to algorithmically perform pattern discovery. If processed effectively we will then be able to visualize clusters; learn nested structure in the data that may be contrasted with theoretical models; distinguish between various acoustic groupings and classify unseen data into their instruments with great precision.

These techniques were applied on the audio signals produced by a range of instruments. First digital signal processing techniques such as the short time fourier transform and mel filter banks were used to find a reduced representation in the frequency domain. Both linear and kernel PCA were explored for data visualization - with kernel PCA capturing 54% of the variation with two components - however these methods were not effective in reducing the data in a meaningful and interpretable way.

K-means, Gaussian mixture models and hierarchical clustering were used to attempt to find the groupings of different timbre, but instead found the notes within the signals - an intuitive higher broader grouping. A more directed approach to differentiate between the instruments was taken using a convolutional neural network, with the purpose being classification. These networks were able to pick up the timbre, having a 95.45% accuracy.

2 Literature Review

2.1 A numerical representation of audio

Sound can be described as the transfer of movement, through movement pressure vibrations are created and transmitted as a result. These vibrations are then picked up by our eardrums, which essentially act as a transducer transforming the vibrations into nerve impulses which our brain interprets as sound. A visual representation of these vibrations can be seen as a waveform in Figure 2. Mathematically, this waveform can be interpreted as a function of time, or simple an image with varying pixel densities.

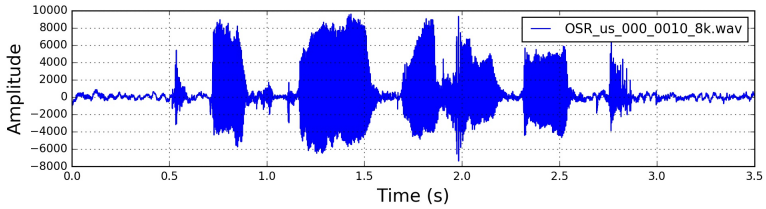


Figure 2: Example of an audio signal [1]

This waveform encapsulates all of the data needed in order to recreate the sound in its entirety. The amplitude indicates the volume of the sound. Different instruments create slightly different sounds even when creating the same note (frequency), this phenomenon is known as timbre - or music colour - and is represented by the slight fluctuations on the waveform inherent to that instrument. Bass can be added by increasing the amplitude of the low frequencies, whilst conversely the same can be done for treble by increasing the amplitude of the high frequencies.

2.2 Frequency Domain

Although the waveform hosts all information, it is impractical to perform statistical techniques on either representation of the waveform (the function or the image) as a consequence of the sparse high-dimensional nature of the data; or the impracticality of true functional approximation; a lower-dimensional rendering is essential.

Audio waves are made up of a combination of multiple sinusoidal signals with varying frequencies, transforming the audio wave into the frequency domain can in turn reduce the complexity of the problem. The process of transforming a signal from the time-domain to the frequency-domain is called the fourier transform. There are different implementations of this transform, each with it's own trade-offs. The choice of implementation will depend on the stationarity of the signal.

2.2.1 Fast Fourier Transform

In order to capture a continuous audio signal in the time domain, the signal is sampled at finite intervals known as the sampling rate, the higher the sampling rate the higher the frequencies that can can be retained. The process of converting these samples into the discrete frequency domain is called a DFT (Discrete Fourier Transform), computing the DFT can be computationally expensive and so the fast fourier transform - an algorithm that efficiently computes the DFT allowing for faster computation - is used. This method does not account for changes in frequency over time, and works best on stationary signals. If it is performed on a non-stationary signal, information will be lost when transforming back into the time domain.

The equations for computing the fast fourier transform and inverse fast fourier transform are expressed below

$$F(x) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi(x)\frac{n}{N}} \tag{1}$$

$$f(n) = \frac{1}{N} \sum_{n=0}^{N-1} F(x) e^{j2\pi(x\frac{n}{N})} \tag{2}$$

2.2.2 Short-Time Fourier Transform

The Short-Time Fourier Transform is a fourier related transform that calculates the fourier transform at discrete time intervals, allowing for the non-stationarity of a signal to be captured. These discrete times are captured by a window function which captures the discrete intervals in the time domain by zero-padding the rest of the signal (setting the time samples outside of the time interval to zero), the fft is then applied to this windowed signal.

2.2.3 Mel Filterbank

For a human observer, frequency discrepancies at higher frequencies are far harder to distinguish than the equivalent absolute change in lower frequencies. The Mel filterbank takes this into consideration by aggregating frequencies on a log scale - the higher the frequencies the larger the range that gets averaged together. The Mel filterbank can be viewed in Figure 9. Since frequencies are combined, the filterbank helps reduce the dimensionality in the frequency domain.

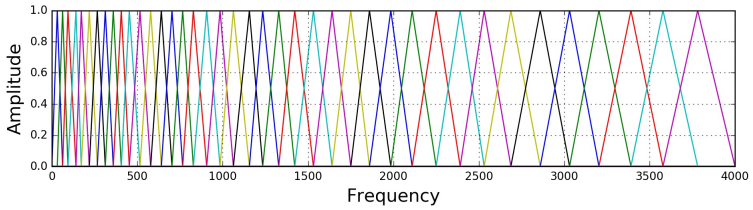


Figure 3: Example of a Mel filterbank [1]

The equations to compute between the frequencies f and the Mel coefficients m are as follows:-

$$m = 2595\ log_{10}(1 + \frac{f}{700}) \tag{3}$$

$$f = 700(10^{\frac{m}{2595}} - 1) \tag{4}$$

2.3 Dimensionality reduction techniques used for images

By visualizing the change in frequencies over time as an image in a spectrogram (a representation of the spectrum of frequencies as they vary over time), we can reduce the dimensionality of the original audio signal. For example, a spectrogram after a mel filterbank has been applied to the discrete time frequency spectrums of the audio signal can be seen in Figure 4. However, images are still considered as high dimensional and further techniques are needed in order to further reduce the data. Two promising techniques which have been researched for this application are: PCA [4] and Kernel PCA. The comparison of the two should allow for the discovery of both linearity and non-linearity in the data.

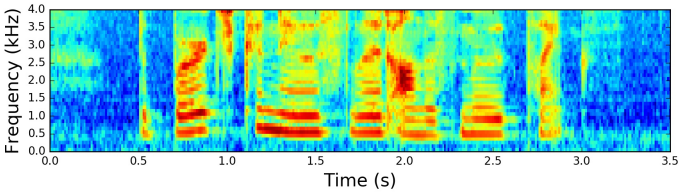


Figure 4: Example of spectrogram after a Mel filterbank [1]

PCA is explored as a technique for reducing the dimension of an image in a paper by Sok Choo, Ng [4]. The paper shows how PCA can be used in order to find principle patterns within the images, allowing the image to be reduced without losing it's core information. The results are promising and since a spectrogram can be viewed as an image, this shows potential as an application for this problem.

2.4 Clustering and classification

With a lower dimensional representation of the data, the task of clustering similar audio waves becomes achievable. Multivariate methods such as k-means, Gaussian mixture models and hierarchical clustering can be used in order to find these similarities. Hewage has shown the practicality of clustering the principle components of images in his paper on clustering colour images for screen-printing [3].

The purpose of these clusters is to be able to understand the predominant features of the signals that make them similar/dissimilar.

Convolutional neural networks are used to pick up patterns within images, these networks can be used for the purpose of classifying between different images. Since we are able to reduce our data into an image of frequency over time, these networks make a perfect use case for differentiating between audio signals.

Traditional neural networks negate temporal dependencies and simply learn non-linear relationships in the data. Recurrent nets are a class of algorithms that attempt to discover temporal relationships in the data by tracking state [2]. This proves useful in scenarios where sequences of inputs are largely dependent on the preceding sequence - as in the case of text or audio data and thus also make for a good use case for differentiating between audio signals.

3 Understanding the data

3.1 Raw data

The data explored in this assignment is comprised of short audio files of different instruments. There are 30 audio files for each of 10 different instrument types:

Instrument	Instrument family
Saxophone	Woodwind
Violin or Fiddle	String
Hi-Hat	Percussion
Snare Drum	Percussion
Acoustic Guitar	String
Double Bass	String
Cello	String
Bass Drum	Percussion
Flute	Woodwind
Clarinet	Woodwind

Since different instruments can produce similar notes (frequencies) distinguishing between the instruments will depend upon trying to interpret the instrument's timbre. Timbre is dependent on how an instrument produces a note and so it would make sense to look for correlations between instrument families. Differences between the instruments and their respective families will be considered using both the acoustic characteristics of the instruments and audio signals produced by the instruments.

The acoustic characteristics considered were the instrument's tonal range, typical max Sound Pressure Level (SPL) at 3m and attack time. The tonal range refers to the range of frequencies that the instrument can produce. The max SPL at 3m is the maximum sound pressure at 3m that can be produced by the instrument and the attack time refers to the time it takes for a wave form to get from 0 to it's maximum level.

The audio waves all differ in length and in order to create more data to train the neural networks on the audio waves are split into 0.1 second intervals. And so, although the number of files per instrument is consistent, the split data hosts unbalanced classes as the files vary in length. The average length of the samples of each instrument can be seen in Figure 5. Splitting the data into 0.1 second intervals resulted in 26410 observations which can be used for the purpose of better generalization when fitting models.

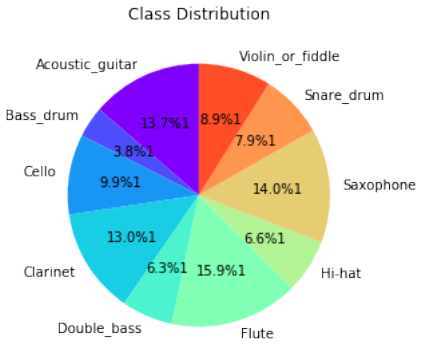


Figure 5: Class imbalances in the data

The magnitudes of the frequencies die down over time leaving instances were there is not much of a signal, this can be viewed in the left graph of Figure 6. These areas of the signal with low magnitudes are hard to interpret and so noise threshold detection is used by applying an envelope function and removing any part of the signal below a certain magnitude threshold. The transformed waveform can be viewed in the right hand graph, were all information within the waveform is relevant.

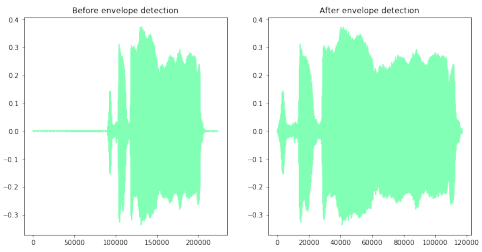


Figure 6: Noise threshold detection

Once the irrelevant data had been removed from the signals, the signal's sampling rates were reduced from 44100 kHz to 16000 kHz through a process called down-sampling. Since we are working with instruments with tonal ranges below 16000 kHz the higher sampling rate just increases the size of the data unnecessarily.

3.2 Frequency Domain

The original data is in the amplitude-time domain as can be visualized in Figure 7. These waveforms are difficult to work with, since when viewed as images provide extremely sparse and similar representations for distinctly different sounds.

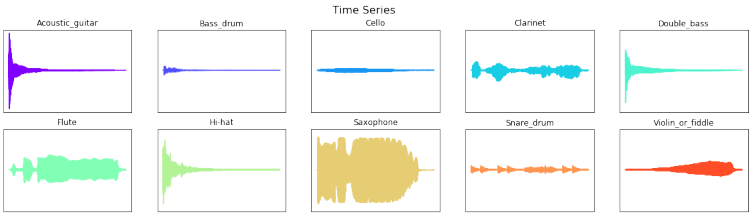


Figure 7: Unprocessed audio data given in wave form

Transforming the audio waves into the amplitude - frequency domain allows us to visualise the predominant frequencies present in each signal. The frequencies can be visualised in Figure 8. All audio signals of the instruments are comprised of mainly low frequencies. This is further evidence that the downsampling didn't lose any important information. The hi-hat and snare drum have frequencies across a larger range of the frequency spectrum and so it is hard to distinguish between the instruments from the ffts.

Figure 8: Applying the Fourier Transform to the data displays the commonality of frequencies for each audio file

To account for the changing frequencies over time we looked into a spectrogram (a representation of the spectrum of frequencies as they vary over time) over 1 second as can be seen in Figure 9. The differences between the instruments become more apparent and the temporal aspect can be visualized.

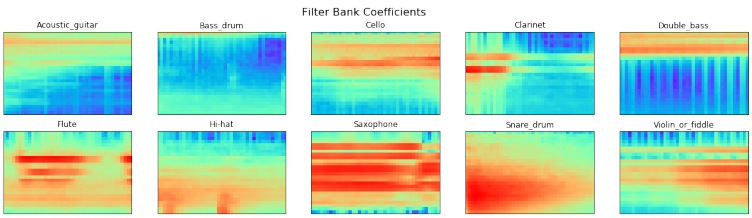


Figure 9: Applying the short-time fourier transform that results in discretely binned frequencies that begin to depict patterns in frequency densities overtime

A Mel filterbank reduced our frequencies into 26 mel frequency cepstrum coefficients. Since our data has frequencies predominantly in the lower frequency

band we only kept the first 13 coefficients. The averaging over the frequency bands reduces the temporal component of the data as it stops picking up smaller fluctuations over time and the signals start looking more stationary.

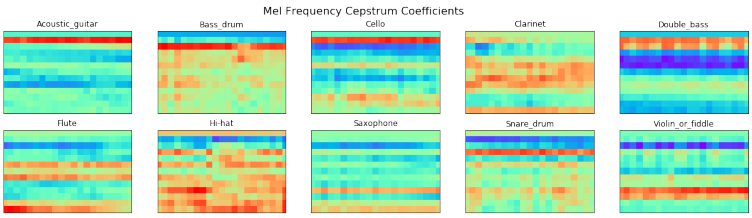


Figure 10: Processed data after applying a Mel Filterbank

4 Multi-Dimensional Scaling

All instruments have their own set of acoustical characteristics such as it's tonal range, typical max SPL at 3m and attack time.

These acoustic characteristics were used to create a distance matrix of the musical instruments, multi-dimensional scaling was then used to visualize the distances between the instruments. K-means was used to cluster the instruments based on the acoustic characteristics.

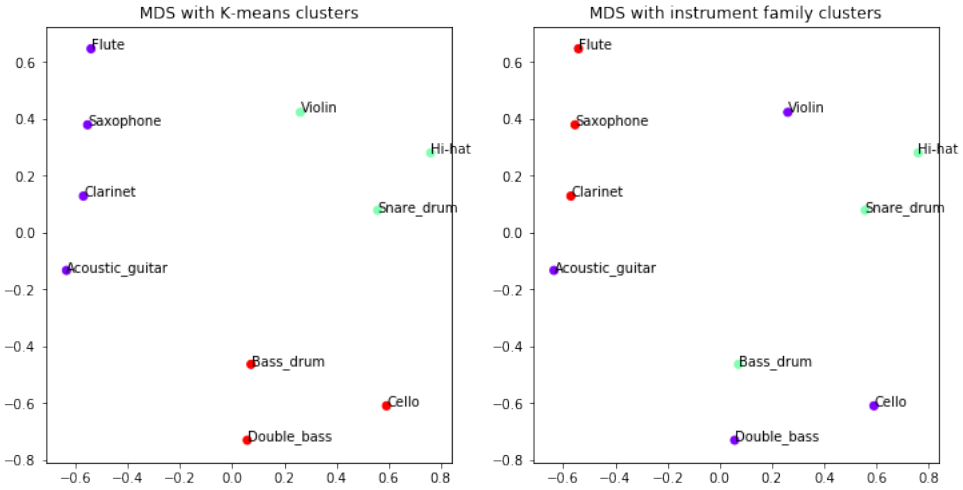


Figure 11: Multi-Dimensional scaling on the acoustic characteristics

Figure 11 shows the mds mapping of the points coloured based on their k-means cluster labels on the left and instrument family on the right. The clusters

found from acoustic characteristics showed overlap between the different instrument families indicating overlapping acoustic characteristics. The string family indicated in purple has the greatest variation across and hence the greatest variation on acoustic characteristics.

The clusters found based on acoustic characteristics and families of instruments will be taken into account throughout the rest of the techniques applied.

5 Dimensionality Reduction

Using the mel cepstrum coefficients we were able to reduce our data drastically to a 13 by 9 matrix for a 0.1 second snippet of an audio signal. Despite this huge reduction we are still left with 117 variables.

In the previous section we saw that the mel coefficients remained relatively stationary over time and so we conducted dimensionality reduction of both the 13 x 9 matrix as well as just a 13 x 1 vector of the data with the assumption that the coefficients are not changing over time. The 13 x 9 matrix was flattened into a vector of 117 variables and the following dimensionality reduction techniques were applied. Less variance was able to be captured on the 117 variables and it made the visualization less intuitive. For this reason the following techniques are utilized on the 13 x 1 variable vector.

5.1 PCA

Principle component analysis was applied to the 13 variables to see if any linear correlations between the variables could be found and in turn be able to explain the same amount of information in fewer dimensions. Figure 12 shows the scree plot of the variation explained by each of the components. The first two principle components were able to explain 44 % of the variance in the data, with majority(96.34%) of the data captured by 12 components. This unfortunately doesn't allow for much dimensionality reduction. The following subsection looks into kernal PCA a non-linear dimensionality reduction technique to see if non-linear correlations can be found.

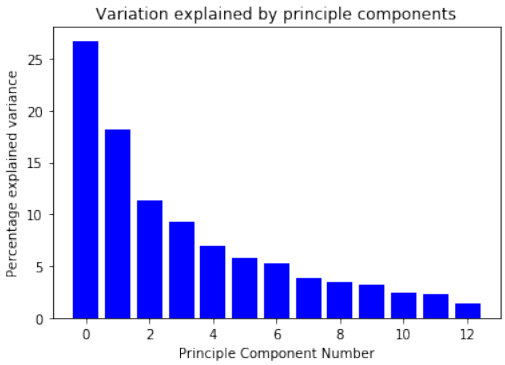


Figure 12: Linear PCA scree plot

Figure 13 shows the mapping of the first two principle components, the further left graph is coloured by instrument - no clusters emerge and the distribution of instruments appears random. This gives indication that the first two principle components do not hold enough information to differentiate between the instruments. The central and right graphs map colours on the first two linear PCA components by acoustic characteristics and instrument families respectively. Though considerable cluster overlap, more coherent groupings are found - perhaps a consequence of PCA capturing the most notable differences in the data (larger hierarchical constructs) whilst unable to adequately learn nuance differences between instruments.

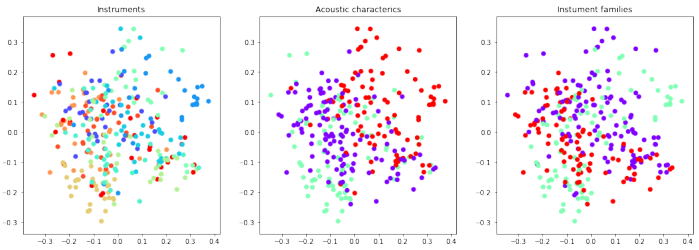


Figure 13: Known groupings visualised on linear PCA components

Since the first two principle components failed to capture enough information about the differences between instruments, it was further explored to see what we could extract from them. Figure 14 shows the points coloured by their predominant frequency value. The gradient changes from bottom left to top right with higher frequencies shown in lighter colours and lower frequencies shown in purple and blue. And so even though we can't extract information about the audio signal's timbre we can are able to extract information about the audio signal's note.

When comparing the frequency values to the acoustic characteristics and family clusters it appears that the acoustic characteristics align with the frequency values. Further, showing that the string family indicated by purple and the woodwind indicated by red families seem to have differing frequencies whilst the percussion family indicated by green completely overlaps the other two families.

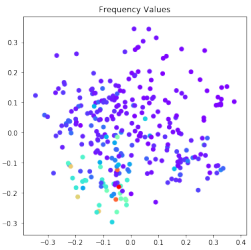


Figure 14: Predominant frequency values visualised on linear PCA components

5.2 Kernal PCA

Figure 15 shows the scree plot of the variation explained by each of the components produced by kernel PCA using a polynomial kernel function of degree 6. The first two principle components were able to explain 53 % of the variance in the data, with majority (96.8%) of the data captured by 12 components. Although this method also doesn't allow for much dimensionality reduction the first two components capture more variance than linear PCA and so the kernel PCA's first two components will be used for the purpose of visualization throughout the rest of the paper.

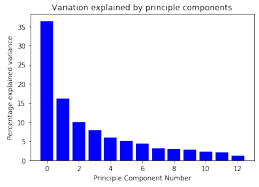


Figure 15: Scree plot for kernel PCA

Looking at the known groupings mapped onto the first two principle components, the kernel PCA has not helped provide more distinct clusters. This makes sense as only an additional 9% of the variance is explained.

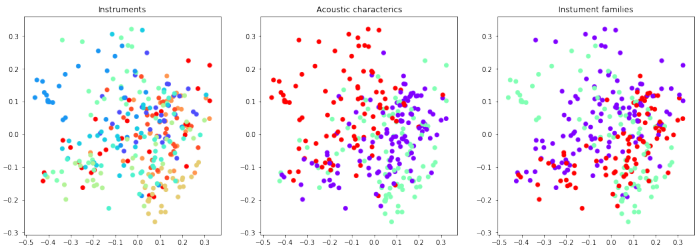


Figure 16: Known groupings visualised on kernel PCA components

Grouping buy frequencies is still a gradient as seen in linear PCA.

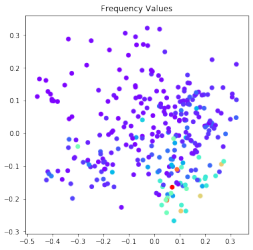


Figure 17: Predominant frequency values visualised on kernel PCA components

6 Clustering

Clustering methods were explored on all the 13 variables since the dimensionality reduction techniques failed to find a reduced mapping that could retain sufficient information. These clusters were then plotted onto the first two principle components of the kernel PCA in order to visualize the groupings found by the techniques and contrast them to our known theoretcal groupings.

6.1 K-means

The gap statistic, sillhoutte distance and within sum of squared distances were used to try determine the optimal number of clusters using k-means. Figure 18 shows the method's values for a range of cluster values. These graphs point towards there being 3 clusters within the data, although the values are quite low showing it was struggling to find distinct clusters. This could be expected as the clusters we are attempting to find severely overlap.

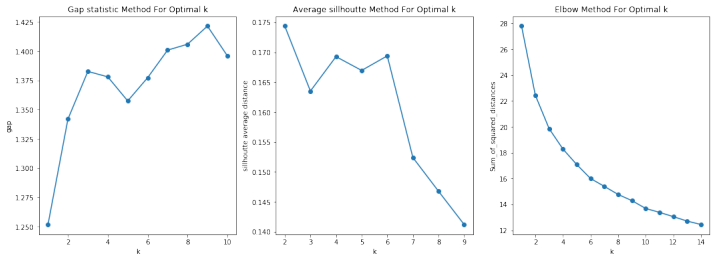


Figure 18: Different methods for finding the optimal number of K-means clusters

Figure 19 shows the k-means clustering of the data, which is unable to approximate the known groupings in the data.

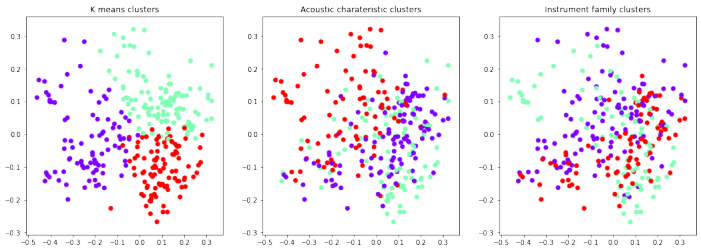


Figure 19: K-means clustering and known groupings visualised on kernel PCA components

When comparing the k-means clustering to the frequencies of the data, it seems to pick up the high frequency grouping whilst mis-classifying some of the lower frequencies into the cluster, this is due to the nature of k-means as it finds globular clusters within data. The lower frequencies seem to have been split into two groups.

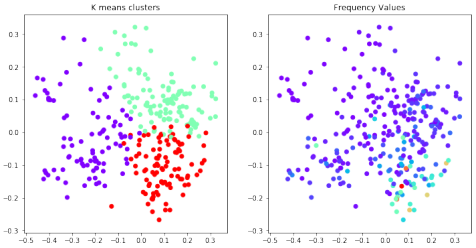


Figure 20: K-means clustering and predominant frequencies visualised on kernel PCA components

6.2 Gaussian Mixture Model

The gap statistic, sillhoutte method and wss elbow method give conflicting feedback on the number of clusters gmms find in the data. The gap statistic suggests 3 clusters and because our known groupings contain 3 clusters, 3 clusters were learnt.

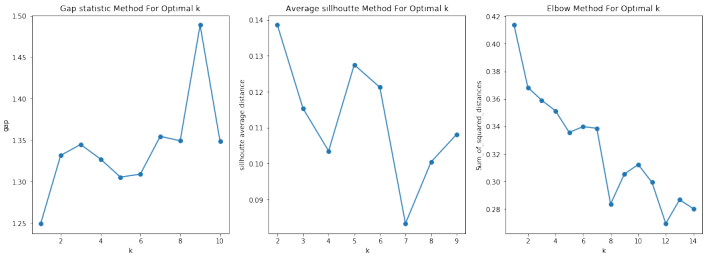


Figure 21: Different methods for finding the optimal number of clusters using GMMs

Figure 22 shows the clusters found by the gaussian mixture model. Although it fails to capture the acoustic characteristics of family clusters, it does allow for overlap which k-means failed to do.

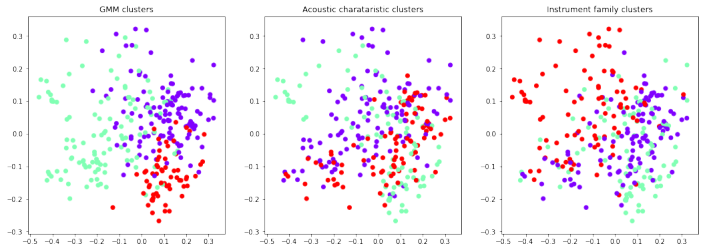


Figure 22: GMM clustering and known groupings visualised on kernel PCA components

Since there is an allowance for overlap due to the soft-clustering nature of the algorithm, the found clusters are able to capture the high frequencies without classifying the low frequencies in the same area into the same cluster. It too finds two clusters within the low frequencies, these clusters split the data across the negative diagonal, whilst the lower frequencies seem to split across the positive diagonal according to their frequency. These clustering algorithms must be differentiating these frequencies according other predominant frequencies with the signals.

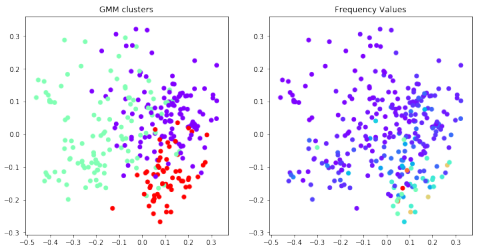


Figure 23: GMM clustering and predominant frequencies visualised on kernel PCA components

6.3 Hierarchical clustering

Again optimal cluster size heuristics yield contradictory results, but for convenience we use $K = 3$ as it indicated by the gap statistic and is the known true structure of the data. Ward linkage was used as it yielded the most coherent results - ward minimizes the variance of the clusters being merged.

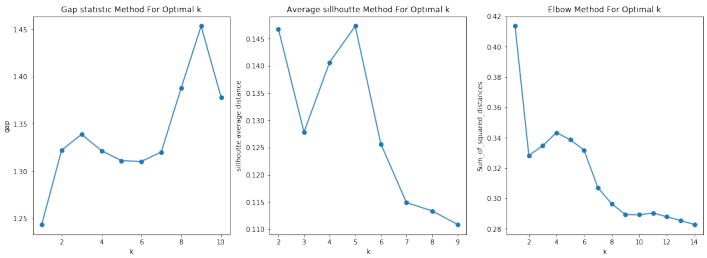


Figure 24: Different methods for finding the optimal number of clusters using hierarchical clustering

Hierarchical Clustering is a logical choice given the considerably nested structure in the data: although audio types are distinct and unique, all instruments belong to families and acoustic groupings. One might expect nontrivial correlations within group members, as such imposing a hierarchical structure may fit soundly.

We hierarchically clustered the data from 1 (all data points allocated to a single cluster) to 10 (1 cluster per instrument) clusters. Coupling this with performing dimensionalilty reduction techniques, we were able to explore visualise agglomerative hierarchical clustering's ability to learn latent constructs. All configurations are available in the appendix, however more salient findings are show below.

Whilst both tSNE and kernel PCA were applied, notable, neither dimensionality reduction technique is able to separate family nor acoustic groupings

in a truly satisfactory manner - yet soft collections do emerge, seen in 25.

The point of interest in these comparisons is how closely the learnt clusters relate to the known underlying latent structure (acoustic and family groupings). Figure 25 utilizes kernel PCA. Although the divide between learnt clusters is inconclusive, we observe that clusters roughly similar to both family and acoustic groupings are modeled with hierarchical clustering - indicative of the highly correlated nature of within group samples.

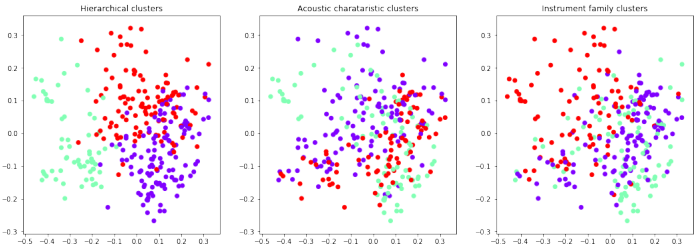


Figure 25: Kernel PCA dimensionality reduction coloured by learnt and known groupings

The purple hierarchical cluster as can be seen in Figure 26 groups both the high and medium frequency values together which was not done by the previous clustering methods. This method found the medium frequencies closer to the higher frequencies, whilst the other clustering methods put them closer to the lower frequencies. The hierarchical clustering found the distinction along the negative diagonal between the lower frequencies as the other clustering methods did.

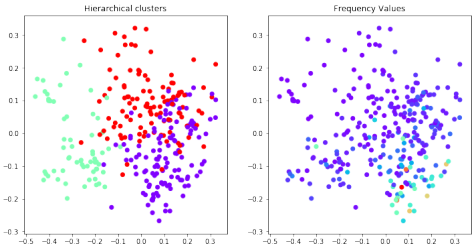


Figure 26: Kernel PCA dimensionality reduction coloured by learnt and frequency groupings

For completeness we ran all clustering algorithms on tSNE. Although it produces apparently random clusters the majority of the time, interestingly tSNE did (on a given hierarchical clustering run) appear to learn the true latent structure of the data, seen in figure 27, with clusters closely resembling the

instrument family groupings. All other tSNE-clustering experiments are in the appendix.

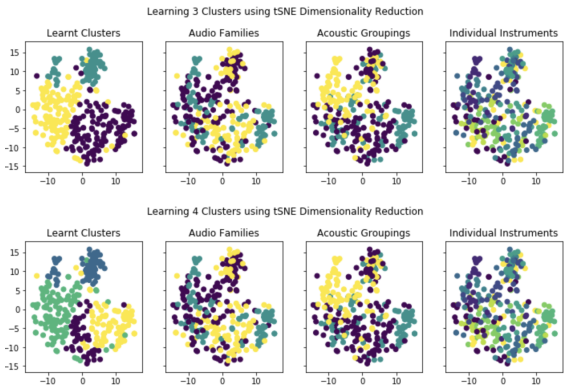


Figure 27: tSNE dimensionality reduction indicating learnt clusters are able to coincide with then known structure of the data

7 Neural network classification

A convolutional neural network was used to classify between the different instruments. Figure 28 shows the train accuracy vs the validation accuracy. From 5 epochs on wards the training accuracy increases whilst the validation accuracy starts to decrease, indicative of over-fitting, so parameter weights were used after 5 epochs.

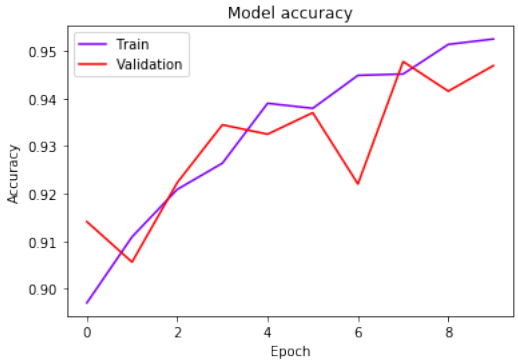


Figure 28: Test vs validation accuracy over multiple epochs

The model was then used to predict the classes of on the test set (1/3 of the data). It classified the test set with a 95.45% accuracy. The predictions can be seen in the confusion matrix below.

	0	1	2	3	4	5	6	7	8	9
Acoustic guitar	1021	6	24	3	15	9	0	0	2	5
Bass Drum	2	158	4	0	4	0	0	0	1	0
Cello	13	3	839	10	47	7	1	13	0	12
Clarinet	0	1	2	1197	0	9	0	2	0	1
Double Bass	9	9	15	3	532	0	0	0	0	1
Flute	13	3	2	17	2	1322	1	0	0	6
Hi-hat	0	0	0	0	0	0	593	0	0	0
Saxophone	1	0	6	14	0	23	0	1309	1	6
Snare Drum	1	0	6	3	4	1	2	1	565	4
Violin	7	0	13	5	3	10	0	9	0	78

A consequence of the inordinate parameterization of Neural networks, coefficients are not readily interpretable. The benefit of this is that it allows for discovery of many non-linearities in the data. The network's superb ability to capture the mapping between inputs and their responses may allude to the non-linear nature of the data.

The diagonals along the confusion matrix denote accurate predictions. The model classified all hi-hat sounds correctly, it's average frequency is far higher than the other instruments - making it a fairly unique sound.

This largest misclassification was predicting a cello to be a double bass. This could be expected because the pair structurally similar - consisting of the same family and acoustic grouping, figure 11 depicts near proximity between the two in a dimensionailty reduction context.

These results were achieved with a convolutional neural network - ordinarily used to learn the correlated mapping of pixel densities in image. This approach negates temporal dependencies in each image. As such we contrasted these findings with those learnt with a recurrent neural network (RNN) - which is specified to compute non-linear dependencies whilst incorporating using the prior 'state' (temporal location) as a predictor to the next 'state'. We found that while the RNN yields strong predictive performance, it is inferior to the CNN - possibly a byproduct of the stationarity of the data.

Since the networks are supervised methods and trained for the purpose of instrument classification they are directed to pick up discrepancies - captured in timbre - of each instrument rather than the note. The unsupervised techniques readily learnt differences in frequencies (notes) rather than the more subtle yet still significant timbre.

8 Conclusion

As recommendations for future work, using the digital signal processing techniques predominant frequency component can be extracted from a signal. Using this fundamental frequency, the overtones can be be extracted to find the full representation of the predominant note. These overtones can be used to determine a function $f(instrument, overtones)$ that captures how each instruments creates the different overtones (by their magnitudes). Once notes can be extracted and this function can be found, the classification of instruments becomes a much simpler one.

Further, the empirical decomposition of music can be internalized by musicians to test assumptions about the structure of different compositions - potentially offering unique new tools to aid in the creative, musical, process.

References

[1] Haytham Fayek. Speech processing for machine learning: Filter banks, mel-frequency cepstral coefficients (mfccs) and what's in-between, 2016.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[3] R.U. Hewage and Upul Sonnadara. Colour image segmentation technique for screen printing. 03 2011.

[4] Sok Choo Ng. Principal component analysis to reduce dimension on digital image. *Procedia Computer Science*, 111:113–119, 12 2017.

9 Appendix

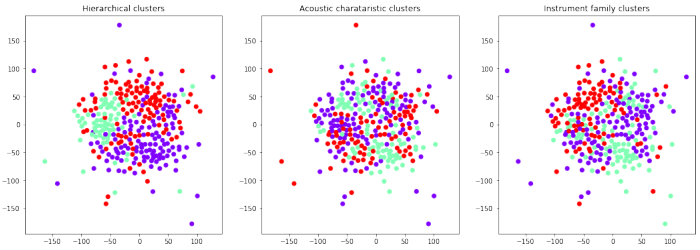


Figure 29: tSNE dimensionality reduction coloured by learnt and known groupings

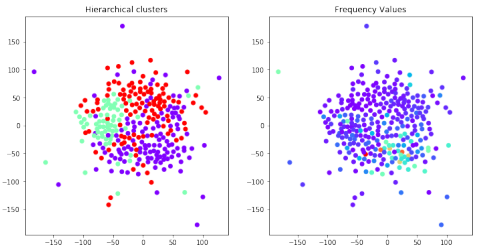


Figure 30: tSNE dimensionality reduction coloured by learnt and frequency groupings

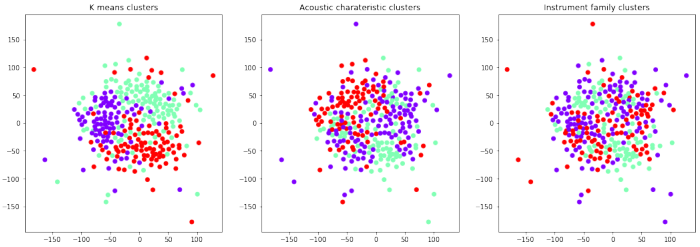


Figure 31: tSNE dimensionality by Kmeans clustering

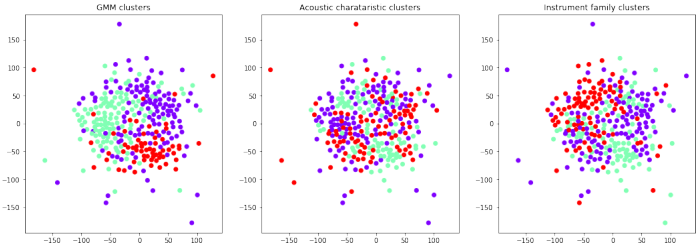
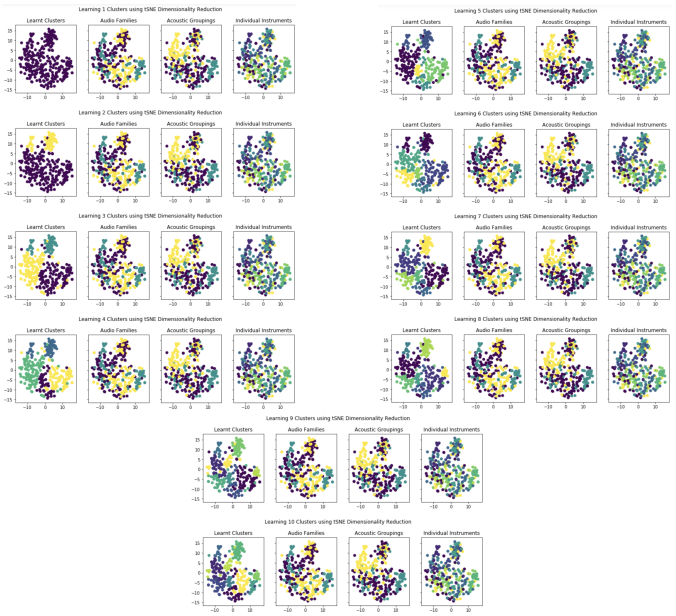
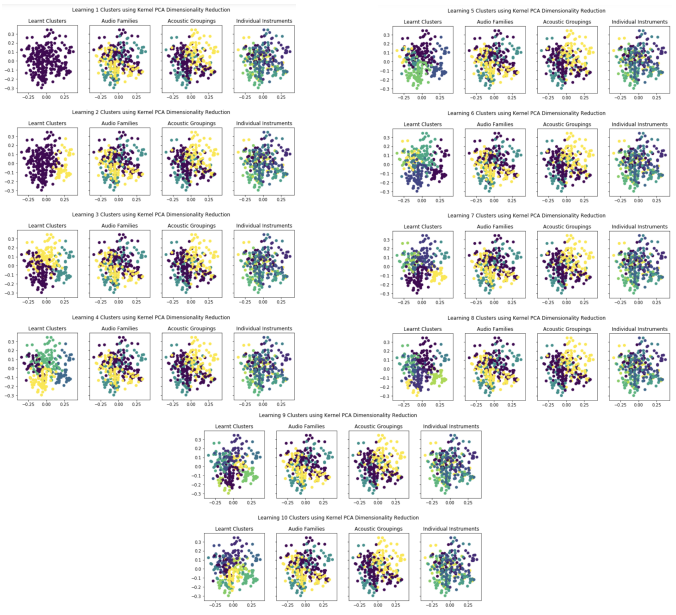


Figure 32: tSNE dimensionality by GMM clustering


Learning 1 Clusters using tSNE Dimensionality Reduction
Learning 2 Clusters using tSNE Dimensionality Reduction
Learning 3 Clusters using tSNE Dimensionality Reduction
Learning 4 Clusters using tSNE Dimensionality Reduction
Learning 5 Clusters using tSNE Dimensionality Reduction
Learning 6 Clusters using tSNE Dimensionality Reduction
Learning 7 Clusters using tSNE Dimensionality Reduction
Learning 8 Clusters using tSNE Dimensionality Reduction
Learning 9 Clusters using tSNE Dimensionality Reduction
Learning 10 Clusters using tSNE Dimensionality Reduction
Learnt Clusters
Audio Families
Acoustic Groupings
Individual Instruments


Learning 1 Clusters using Kernel PCA Dimensionality Reduction
Learning 2 Clusters using Kernel PCA Dimensionality Reduction
Learning 3 Clusters using Kernel PCA Dimensionality Reduction
Learning 4 Clusters using Kernel PCA Dimensionality Reduction
Learning 5 Clusters using Kernel PCA Dimensionality Reduction
Learning 6 Clusters using Kernel PCA Dimensionality Reduction
Learning 7 Clusters using Kernel PCA Dimensionality Reduction
Learning 8 Clusters using Kernel PCA Dimensionality Reduction
Learning 9 Clusters using Kernel PCA Dimensionality Reduction
Learning 10 Clusters using Kernel PCA Dimensionality Reduction
Learnt Clusters
Audio Families
Acoustic Groupings
Individual Instruments