MUGEC: Automatic Music Genre Classification

Hrishikesh Deshpande, Unjung Nam, Rohit Singh
Stanford University
{hrd,unjung,rohitsi}@stanford.edu

June 14, 2001

Abstract

With the huge increase in the availability of digital music, it has become more important to automate the task of querying a database of musical pieces. At the same time, a computational solution of this task might give us an insight into how humans perceive and classify music. In this report, we discuss our attempts to classify to music into three broad categories: rock, classical and jazz. We discuss the feature extraction process and the particular choice of features that we used- spectrograms and mel scaled cepstral coefficients (MFCC). We use the texture-of-texture models to generate feature vectors out of these. Together, these features are capable of capturing the frequency-power profile of the sound as the song proceeds. Finally, we attempt to classify the generated data using a variety of classifiers. we discuss our results and the inferences that can be drawn from them.

1 Introduction

As the amount of multimedia information stored on computer systems has increased, the ability to find information has become more difficult. Also, handling the new multimedia data types brings new challenges to traditional database management system. For example, in a multimedia database it is reasonable and natural to ask for sounds that are somehow "similar to" some represented sound. Multimedia information is commonly classified and retrieved in a manual process, using text descriptions. While such descriptions are satisfactory for some media types, they are often highly subjective, inaccurate or misleading when describing audio. The general problem of Content Based Classification and Retrieval addresses this problem by describing audio information using characteristics such as pitch, loudness or timbre rather than manual text descriptions. Removing this human involvement would allow the development of automatic analysis, segmentation, indexing and retrieval of audio.

From a psychological perspective also, such a study will be very useful. While we have gained significant amounts of empirical knowledge about human perception of sound, there are a lot of gaps in our understanding of how sound, as a physical signal, is mapped into music, a perceptual signal. We also do not understand which physical 'features' are responsible for our perception about the kind of music being played. In this report, we discuss an approach to classifying music, purely by computational methods. We formulate the problem as a supervised machine learning problem. Thus, we can take advantage of the numerous classification techniques available in machine learning. However, the choice of the correct features, a critical choice, is far more difficult to make. In this case, we take our cue from the domain of acoustics.

The report is organized as follows. In the next section, we formulate the problem and discuss some previous work. One of the authors (Unjung Nam) has done some previous work on this topic. We discuss that and other related work. After that, we provide a discussion of the general methodology and a description of our data. The next section deals with the appropriate choice of features for the extracting information out of the songs. After that, we discuss the classification techniques used and the results we got. Finally, we evaluate the results and also look into the psychological plausibility of the approach.

2 Literature Overview

Recent work in this field has usually been based on extracting statistical measures from the sound signal and using that for classification. Wold et al.[WBKW96] divided audio content into 10 groups: animal, bells, crowds, laughter, machine, instrument, male speech, female speech, telephone, and water. Furthermore, instrument sound is classified into altotrombone, cellobowed, oboe, percussion, tubularbells, violin-bowed, and violinpizz. To characterize the difference among these audio groups, the authors used mean, variance, and auto-correlation of loudness, pitch, brightness (i.e. frequency centroid), and bandwidth as audio features. A nearest neighbor classifier based on a weighted Euclidean distance measure was employed. The classification accuracy is about 81% over an audio database with 400 sound files. Another interesting work related to general audio content classification is by Zhang and Kuo [ZK99a],[ZK99b]. They explored five kinds of audio features: energy, ZCR, fundamental frequency, Timber, and rhythm. A lot of previous work in classification of music has been usually done by looking at musical symbols, not audio signals. Computing musicology community has presented several tools for retrieving musical information from the database made of symbolic representation of the scores [HS99]. In an attempt to build music analysis systems, researchers in computer music community have tried to extract content by first transcribing the music into symbolic notation and then using music theory to characterize it. Lambrou et al. [LKSS98] attempted to classify music into rock, piano, and jazz. They collected eight first-order and second-order statistical features in the temporal domain as well as three different transform domains: adaptive splitting wavelet transform, logarithmic splitting wavelet transform, and uniform splitting wavelet transform. For features from each domain, four different classifiers were examined. They are minimum distance classifier, K-nearest neighbor distance classifier, least squares minimum distance classifier (LSMDC), and quardrature classifier. An accuracy of 91.67% was achieved under several combinations of feature set and classifiers. The LSMDC was the best classifier for most feature types.

2.1 Previous Experiments

One of the authors, Unjung Nam, [UN2001] had worked on this problem before. She had implemented three feature extractors:

Spectral Centroid The spectral centroid is commonly associated with the measure of the brightness of a sound (Grey and Gordon 1978). This measure is obtained by evaluating the center of gravity using the Fourier transform s frequency and magnitude information.

Short-Time Energy It provides a convenient representation of the amplitude variation over the time. Its change pattern over the time may reveal the rhythm and periodicity nature of the underlying sound.

Zero-Crossing Rate It is a measure of per unit time. In the context of discrete-time signals, a zero crossing is said to occur if successive samples have different signs. The rate at which zero-crossings occur is a simple measure of the frequency content of a signal.

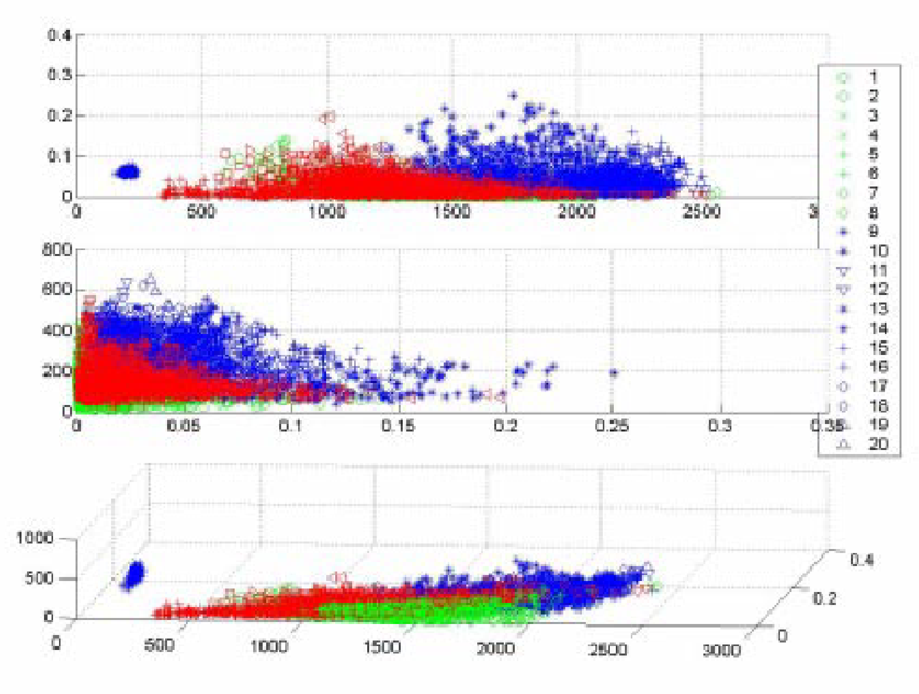


Figure 1: *The red points correspond to classical music, blue ones correspond to pop/rock and green ones to jazz. The first figure shows the spectral centroid along the x-axis and the short-time energy along the y-axis. The second one shows the short-time energy along the x-axis and short-time ZCR along the y-axis. The third figure plots the feature vectors in 3 dimensional space.*

Using these features, a small set of songs (20) was classified. Two main techniques were used, K-Nearest Neighbour classifiers and the K-Means clustering algorithms. For this small set, the classifiers worked very well (about 90%) accuracy. For example, a plot of the various features is shown in Fig 2.1 However, the two main reasons for the success seem to be the very representative training data and the small number of the training and test data.

3 General Methodology

As mentioned above, we had formulated the problem as a supervised machine learning problem. In general, such an approach consists of mapping the training data into feature vectors. One or more classification techniques are applied on this data and a model for the distribution underlying the data is created. Finally, this model is used to estimate the likelihood of a particular category given the test data.

In our case, the procedure can be described as shown in Fig 3. Here is a short description of the main steps in the procedure:

Audio Signal We collected 157 song samples from the internet. From each of those, a 20 second long clip was extracted. These 20 sec long clips were used throughout, both for training and testing.

Feature Extraction From each of these song clips, we extracted various features. This is de-

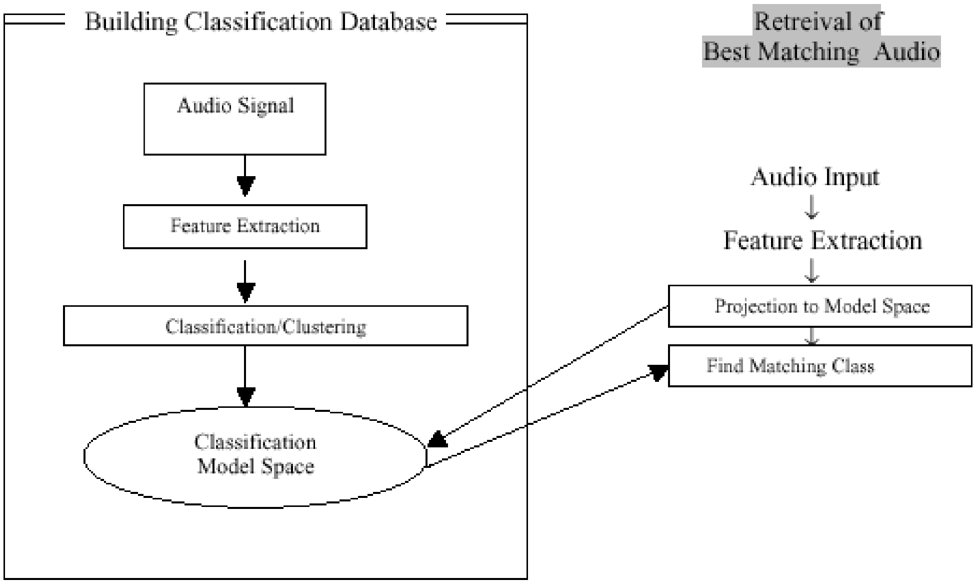


Figure 2: *The procedure we used to classify music samples into the three broad categories: rock, jazz and classical. The process consists of two parts. In the training step, shown in the left box, the data is processed and a model is generated so as to maximize the likelihood of the data given the model. In the testing phase, (after the same preprocessing), the model parameters are used to estimate the category from which the music sample came.*

scribed in detail later.

Classification Once feature vectors had been generated from these music clips, these were fed into classifiers and models for the underlying distribution were generated

Categorization Once generated, these models were used to classify new songs into one of the three categories.

3.1 Collection and Preprocessing of the Audio Signals

We collected our data from the internet. Most of the music samples were downloaded from *http://www.mp3.com*. We chose to download only labeled songs from the website and used these labels to assign categories to the songs. A big problem in the computational analysis of audio signals is that the amount of data is overwhelming. At the same time, we want to use a sufficiently long segment of each song (10-20 secs) in our analysis so as to preserve the basic flavor of the music in the song. As such, it was necessary to downsample the data. However, we wanted to maintain the perceptual properties of the music during this transformation.

The approach we followed was to use MP3 compression to preprocess the data. It is generally agreed that the lossy-compression of the MP3 format nevertheless preserves the perceptual quality of the music ('CD like quality'). Hence, this audio signal would show high variances in perceptually irrelevant features and so would be better for our use than the original CD-audio.

After downsampling the MP3 from 44Khz to 11Khz, we randomly chose a 20 second clip of the song. Such a approach means that, at times, we might capture the song at the 'wrong' moment. Still, choosing the sampling interval randomly seemed to be the best approach.

3.2 Description of our data

We had a set of 157 songs. Of these 52 were rock songs, 53 were from the classical category and 52 were labelled as jazz. Within each category, we took care to introduce sufficient variation. In classical samples, we included samples that corresponded to opera, piano, symphony and chamber music. Similarly, in jazz, we made sure that the songs had a sufficient variety - vocal, fusion, bebop and traditional.

4 Feature Extraction

In an application such as ours, where we need to provide a distance metric between two objects (music clips) which are not directly comparable, we must transform the data into a feature space where we can in fact propose such a metric. Although the metric itself is given by the classifier used, it is defined on the feature space. Since we want our metric to be perceptually meaningful, the choice of features is critical:

1. Objects that map to nearby points in the feature space must in fact be objects that we regard as similar. Hence, for our purpose, we must try to find a feature space where all samples belonging to a particular category (Rock, Jazz, Classical) must cluster closely. At the same time, clusters corresponding to different categories must have a large distance between them. That is, the intra-category scatter must be small whereas the inter-category scatter must be high.

2. Secondly, we want to make sure that the features capture all of the physical knowledge we have of the objects. Then we can be sure that, in theory, we are not missing any information and a well-trained and expressive classifier will be able to do a good job.

4.1 Transforming from the audio to the visual domain

At this stage, we would like to map our audio-classification task to a visual-classification one. The reason for doing this is that there is a promising new approach for feature extraction from images, the Texture-of-Textures approach (described in Section 4.4) proposed by DeBonet and Viola [BV97], that seems to pick out features in an image that are indeed perceptually meaningful. We can make use of this approach if we transform our problem into an image classification task. This is rather easily done, even though we are constrained by the above two criteria. We use the spectrograms (Fourier transform) and Mel-Frequency Cepstral Coefficients to go from the audio to the visual domain.

4.2 Spectrograms

The Fourier transform of a signal, which maps from the time to the frequency domain, is a well known signal processing technique because of good properties such as invertability, linearity, and linear time-invariance. Secondly, frequency has an intuitive meaning, and especially for us, captures an important part of musical information.

Given an audio signal $x(t)$ where t indicates time, we define the short-term Fourier transform at the time t and a frequency f as:

$$X(f,t) = \int_{-\infty}^{\infty} h(t'-t)x(t)e^{-j2\pi ft}dt \tag{1}$$

where $h(t)$ is a window function which is near 1 at $t = 0$ and gradually decreases to 0 on either side, so that we are looking only at the signal through at a small window centered at time t. We use the standard Hann window function which is good at eliminating the edge errors in the Fourier transform.

If we thus choose a suitable width of the time window - the value of $\pm t$ beyond which the Hann window function is zero, we can get, for a frequency bin centered at a certain f, the power that the signal $x(t)$ has in the frequencies corresponding to that bin. We can thus generate a spectrogram by considering all possible time windows, as is shown in the spectrograms of Fig 1. The x-axis shows increasing time t, while the y-axis, starting from top going down, shows increasing frequency f - the frequency bins being of equal size. The color of a point at (f,t) indicates power - going from a value of zero (black) upwards (white). We use a time window size of 512 samples, at a sampling rate of 11025 Hz, with a linear scale to convert from power to the gray-value of the pixel.

We argue that this spectrogram image is a good representation of the audio clip because we can invert a spectrogram to reconstruct the signal, thus we have not lost any of the physical information contained. Secondly, as we see from Fig 1, we see a distinct difference between the characteristics of the spectrograms for the three categories:

- Rock tends to produce strong vertical lines - high power in all frequencies within a short time interval - corresponding to the high transients seen in instruments such as guitars used for rock music. Also seen are characteristic back-quote (‘) shaped curves which correspond to the bends and slides on the guitars.

- Classical tends to be smooth - fading horizontal lines - corresponding to the fact that most classical instruments (piano) produce a pure pitch, which slowly decays in volume across time. The lower part of these spectrograms is almost totally black indicating the absence of high frequencies or transients as in Rock.
- Jazz spectrograms show a huge variation. But if wind instruments have been used then we can see a continuous zig-zag curve corresponding to trimolos.

Thus we see that spectrograms are often visually interpretable, and should be a good way to convert an audio clip to an image.

4.3 Mel-frequency cepstral coefficients

Recently Mel-frequency cepstral coefficients (MFCCs) have been successfully used in distinguishing between speech and non-speech audio signals. This motivated us to investigate their use for music genre classification.

MFCCs can be considered as the results of the following process:

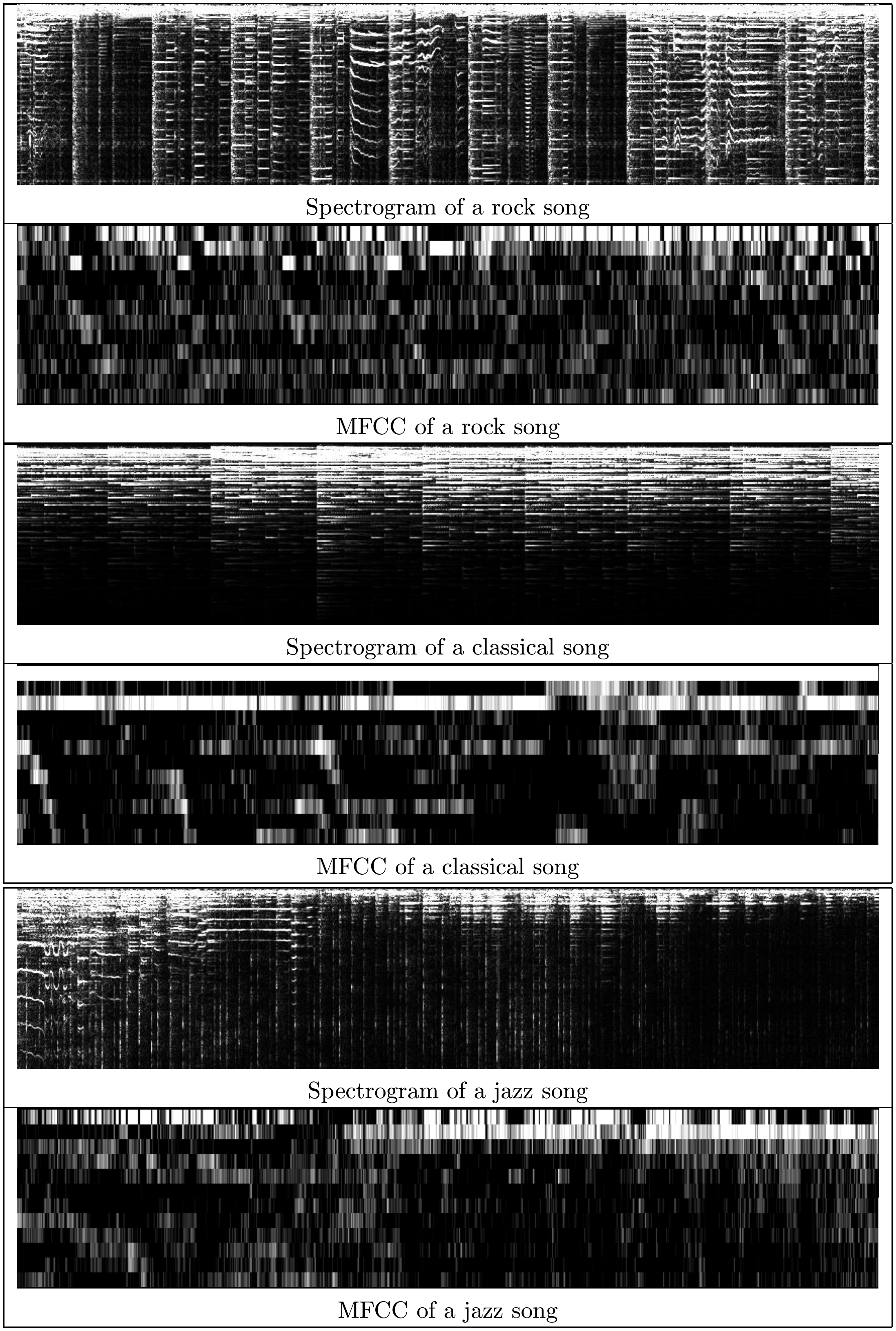
1. Take the short-term Fourier transform of the signal, as above, but instead of dividing the frequency-axis into bins of uniform size, as done in the spectrogram, we divide it according to the Mel-scale, a more acoustically relevant scale. The Mel scale has fixed-size (266 Hz) frequency bins at the lower frequencies, and log-scale sized bins (separated by a factor of 1.07) in the high frequencies
2. We now have about 40 frequency bins. To reduce dimensionality, we perform a DCT on the 40 values (equivalent to a PCA) and get 12 resultant coefficients which are the MFCCs.

Thus, 12 MFCCs are calculated for each time window, and we get a resultant picture as shown in Fig 1, with the same parameters as for the spectrogram. MFCCs are thought to capture the perceptually relevant parts of the auditory spectrum.

4.4 The Texture-of-Textures approach

Now that we have converted from the audio to the visual domain, we can use the recursive texture-of-textures approach proposed by DeBonet and Viola [BV97]. The method uses k filters to operate recursively d times on an image and results in a vector in $\Re^n$ space where $n = k^d$. A summary in follows:

1. An image is convolved with k different filters to result in k different images. In our case, $k = 25$ and these filters represent Gaussians and derivatives of Gaussians oriented in different directions. Thus convolving with these filters would imply that we are either blurring the image or detecting edges oriented in different directions. Each of the resultant images are therefore zero, except at points where the original image has the feature that is being detected by this image.
2. We make the k images positive, by taking absolute values of pixels. (Note: DeBonet takes the square of the value, but we found that for our class of images, that would lead to drowning out of all but few pixels.) We then subsample to reduce image size by half so as to reduce the computational burden as the recursion depth increases.



Spectrogram of a rock song

MFCC of a rock song

Spectrogram of a classical song

MFCC of a classical song

Spectrogram of a jazz song

MFCC of a jazz song

Table 1: *The above figures show images of spectrogram and MFCC data for rock, classic and jazz music.*

3. We now apply the same process to the k images, and continue to do so recursively, till we reach our desired recursion depth d. Doing so means that the new images capture some extremely selective feature. e.g. at recursion depth $d = 2$ we can capture horizontal alignments of vertical edges.

4. We now have k^d images - each of which captures a selective feature. How strongly this feature was present in the original image is indicated by the total power contained in these new images. We therefore sum across all the pixel values in each image to yield a vector of k^d images.

A more detailed overview can be found at
http://www.ai.mit.edu/ jsd/jsd.doit/Research/ImageDatabase/QueryableRepresentations

We tested our classification schemes for recursion depth levels from 1 2 and 3, yielding feature vectors 15, 625 and 15625 elements long, for each of the spectrogram and MFCC images.

Based on these final feature vectors, we are now ready to classify the music.

5 Classification: Methods and Results

We chose to use 17 (randomly selected) songs from each category as training points. The remaining 106 songs were used for validation and testing. Unlike most machine learning problems, in our formulation, the dimensionality of the feature vectors usually exceeds (by far) the available number of data points. Due to high processing time required for each clip, we were restricted in our capability to use more songs for analysis.

5.1 Classification Methods

Given the high dimensionality of the problem, it was hard to visualize the distribution of the data points. As such, we could not pre-decide which technique might be the best. We tried a variety of techniques. A lot of our implementation (in C & Matlab) used publicly available libraries:

K-Nearest Neighbour This technique relies on finding the k nearest training points to the given test point. This approach, though nonparametric, is known to be extremely powerful and there are theoretical proofs that its error is, asymptotically, atmost 2 times the Bayesian error rate. In our case, we used the Euclidean distance metric. We performed calculations for upto 10 nearest neighbours.

Model each category as a Gaussian : If we assume that the underlying distribution for each category is a Gaussian distribution, then we can use the data points to estimate the maximum likelihood values of the parameters (mean and covariance matrix) of the Gaussians. These parameters can then be used to estimate the category of any new test point. Note that we consider only diagonal covariance matrices for easy computation.

Support Vector Machines : SVMs are a technique that rely on projecting the data into a higher dimensional space and looking for a linear separator in that space. Of late, they have found increasing popularity as a classification tool.

5.2 Results

Detailed results are given in the appendix at the end of the paper, but the gist of the results can be summed up as follows:

1. The best 3-way classification accuracy that we got was for KNNs. We managed to get upto about 75% 3-way accuracy.

2. There seemed to be only a weak positive correlation between classification accuracy and increasing recursion depth. The increase in performance in going from recursion depth of one to a depth of two was not matched by the corresponding increase in performance in going from two to three. Intuitively, this could be because the spectrogram and the MFCC images contained relatively simple features that could be inferred even after just one or two levels of recursion. As such, the 3^{rd} level of recursion was probably superfluous.

3. The performance of the classifiers when only spectrogram data was considered was roughly to the performance when only MFCC data was considered. However, when the two were combined, the resulting dataset led to slightly better performance.

4. The Gaussian model never performed really well. This might be indicating that the assumption that the distribution for each category is being generated by a Gaussian is not correct.

5. The SVM was used to get 2-way classifications (i.e *'Rock vs non-Rock'* etc.). SVM gave best results in identifying classical music. It distinguished classical music from non-classical music with a ¿90% accuracy. However, its performance in identifying rock and classical music was not that good. Having observed this, we went back to the KNN results and studied them again. Even KNN did better at classifying classical samples rather than rock or jazz samples.

   Interestingly, SVM's results *degraded* slightly as the dimensionality of the feature vector increased. This can be understood if we realize that SVM blows up the dimensionality by itself and so a very high-dimensional feature vector would probably be blown into 'too-big' a size.

6. Some particular songs were misclassified by all classifiers. Often, jazz pieces which had piano were confused for classical by most of the classifiers.

5.3 More Analysis

The bad performance of the Gaussian model on rock and jazz genres and the excellent performance of the classifiers on classical music led us to suspect that while the datapoints corresponding to classical music were 'neatly clustered', this was not so for jazz or rock music. To confirm this, we tried 2 things:

- We ran the K-means clustering algorithm on the dataset with K=3. It turned out that almost all the classical points were clustered neatly in one cluster. However, both jazz and rock were badly spread out into the three clusters (rock being especially so). This suggested that while there was indeed a single cluster for classical, the same was not true for rock or jazz.

- For each category of music, we did the following: calculate the first 25 eigenvectors of the dataset corresponding to that category. Project **all** the datapoints onto these eigenvectors. Then project these transformed coordinates back to the original feature space. Calculate how much the points in each category have shifted from their original position. The intuition is that for a particular category, if it is well-clustered, the first 25 eigenvectors capture most of the variance. So the difference between the initial location of a datapoint and its final location should not be much. This prediction held out for datapoints belonging to classical music. However, for rock and jazz, this did not happen. As such, our guess became even stronger.

6 Evaluation of the results

The results are reasonably good, but there have been better results in classifying music samples[ZK99a],[LKSS98]. However, we had very few data points, especially considering the high dimensionality of the feature space. As such, it is a valid question to ask if our approach will really scale up and give better performance if more and more training samples are provided. An observation that we made was that, atleast in some cases, the classifiers seemed to be making the 'right' mistakes. There was a song clip that was classified by all classifiers as rock while it had been labelled as classical. When we listened to it, we realized that the clip was the final part of an opera with a significant element of rock in it. As such, even a normal person would also have made such an 'erroneous' classification. As mentioned before, pieces of jazz music which had a high piano component were often confused for classic pieces.

6.1 Plausibility of the Feature Space

The psychological plausibility of applying the texture-of-texture models to audio data is debatable. Human beings seem to be capable of inferring the genre of the music very quickly. But our guess is that the texture-of-texture model needs a song clip of a reasonable duration before it can detect enough features.

At the same time, the texture-of-texture models seems capable of detecting the tempo or the speed of the music, which is probably a significant criterion in most cases. Similarly, most classical instruments have only the fundamental and a few overtones. In contrast, many rock instruments produce transients and sounds all across the auditory spectrum. The spectrogram and the texture-of-texture methods also seem to capture that. This is also a desirable feature.

6.2 Is the problem inherently hard?

Except for classical music, our current classifiers couldn't really find 'neat' clusters for the rock and jazz genres. The performance of a non-parametric method like KNN is much better than the performance of a model-based approach like Gaussian Model. This could mean that either we don't have the correct parameters for the model or that we don't have the correct model. It is possible that for, say, rock there are independent sub-categories (isolated manifolds in the feature space) and hence modeling it with a single Gaussian is bound to fail. The opposing argument can be that classifiers have not been able to estimate the correct parameters. This is certainly plausible given the small number of test points, compared to the dimensionality.

7 Conclusion

In this project we have tried to attempt the classification of music into rock, classical and jazz. We achieved reasonable success, especially in the case of classical music. Our approach has raised many interesting questions on which future work can be done. One would be do an analysis of the variation in how people classify music into different genres. That would provide a good estimate of the difficulty of the problem and a gold standard to benchmark automated classifiers against. Another approach would be to get many more datapoints and see if the performance of our classifiers improves. We would also have liked to try other classification techniques and try to fit different models to the data. This could also be explored further.

References

[HS99] Hewlett, W. B. and E. Selfridge-Field, eds. (1999) *Melodic Similarity: Concepts, Procedures, and Applications* Cambridge, Massachusetts: MIT Press.

[LKSS98] Lambrou, T., P. Kudumakis, R. Speller, M. Sandler, and A. Linney. (1998) *Classification of audio signals using statistical features on time and wavelet transform domains* In International Conference on Acoustics, Speech, and Signal Processing (ICASSP-98), vol. 6, (Seattle, WA), pp. 3621-3624.

[ZK99a] Zhang, Tong and C.-C. Jay Kuo. (1999a) *Hierarchical System for Content-based Audio Classification and Retrieval* In Proceedings of International Conference on Acoustic, Speech, Signal Processing. vol 6. pp. 3001-3004. March.

[ZK99b] Zhang, Tong and C.-C. Jay Kuo. (1999b) *Heuristic Approach for Generic Audio Data Segmentation and Annotation* In Proceedings of 7th ACM on Multimedia. pp. 67-76.

[WBKW96] Wold, E., Thom Blum, Douglas Keislar, and James Wheaton. (1996) *Classification, Search, and retrieval of audio* refined version appeared in IEEE Multimedia 1996, Vol.3, No. 3. p.27-36.

[UN2001] Nam, Unjung (2001) A Short Study on Music Classification *http://www-ccrma.stanford.edu/ unjung/AIR/final4web.pdf*

[BV97] De Bonet, J and Viola, Paul (1997) *Structure Driven Image Database Retrieval* in Advances in Neural Information Processing Vol 10, 1997

Appendix: Results of Classification

Note: Results are only provided for the case where the data from spectrogram and mfcc was combined

Classification using K-Nearest Neighbour on data from spectrogram and MFCC combined:
K = number of nearest neighbours considered
D = dimensionality of each feature vector
51 training points, 106 test points

K	1	2	3	4	5	6	7	8	9	10
D=25+25	0.71	0.61	0.69	0.72	0.68	0.72	0.69	0.70	0.67	0.68
D=625+625	0.68	0.61	0.68	0.70	**0.75**	0.72	0.69	0.69	0.68	0.67
D=15625+15625	0.72	0.64	0.73	**0.74**	0.71	0.72	0.67	0.70	0.68	0.69

Classification using a Gaussian model of data from spectrogram and MFCC combined:
Each category is modelled by a Gaussian with a diagonal covariance matrix:
D = dimensionality of each feature vector
51 training points, 106 test points

D=25+25	**0.72**
D=625+625	0.64
D=15625+15625	0.65

Classification using SVM on data from spectrogram and MFCC combined:
C = soft margin regularization constant = 0.1
D = dimensionality of each feature vector
51 training points, 106 test points

	Rock vs non-Rock	Classic vs non-Classic	Jazz vs non-Jazz
D=25+25	0.7500	0.8750	0.7115
D=625+625	**0.8558**	0.8654	0.7212
D=15625+15625	0.8365	**0.9327**	**0.7981**