CLASSIFICATION OF MUSIC SIGNALS IN THE VISUAL DOMAIN

Hrishikesh Deshpande, Rohit Singh

Computer Science Department

Stanford University
hrd,rohitsi@stanford.edu

Unjung Nam

Music Department
Center for Computer Research in
Music and Acoustics(CCRMA)
Stanford University
unjung@ccrma.stanford.edu

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 paper, we discuss our attempts to classify 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. GENERAL METHODOLOGY

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. The procedure can be described as follows:

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 described 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.

2. 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 labelled songs from the website and used these labels to assign categories to the songs.

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.

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.

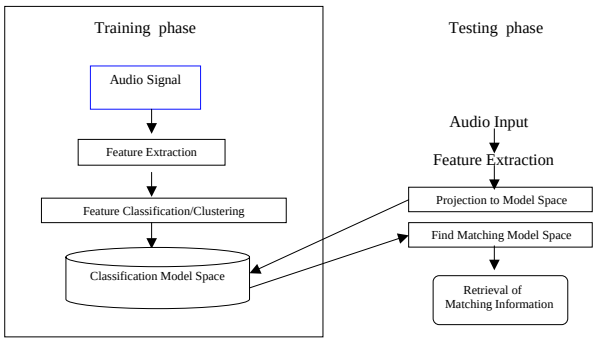

Figure 1: *In the training step, shown in the left box, the data is processed and a model is generated so as to maximize the likelihood of the data given the model. In the testing phase, (after the same preprocessing), the model parameters are used to estimate the category from which the music sample came.*

3. 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.

3.1. Transforming from the audio to the visual domain

At this stage, we map our audio-classification task to a visual-classification one. There is a promising new approach for feature extraction from images, the Texture-of-Textures approach (described in Section 3.4) 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 and Mel-Frequency Cepstral Coefficients(MFCC) to go from the audio to the visual domain.

3.2. Spectrograms

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 the 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 Table 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 tremolos.

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

3.3. Mel-frequency cepstral coefficients

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

1. Take the short-term Fourier transform of the signal, we divide it according to the Mel-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 Table 1, with the same parameters as for the spectrogram. MFCCs are thought to capture the perceptually relevant parts of the auditory spectrum.

3.4. The Texture-of-Texture 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.
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.

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.

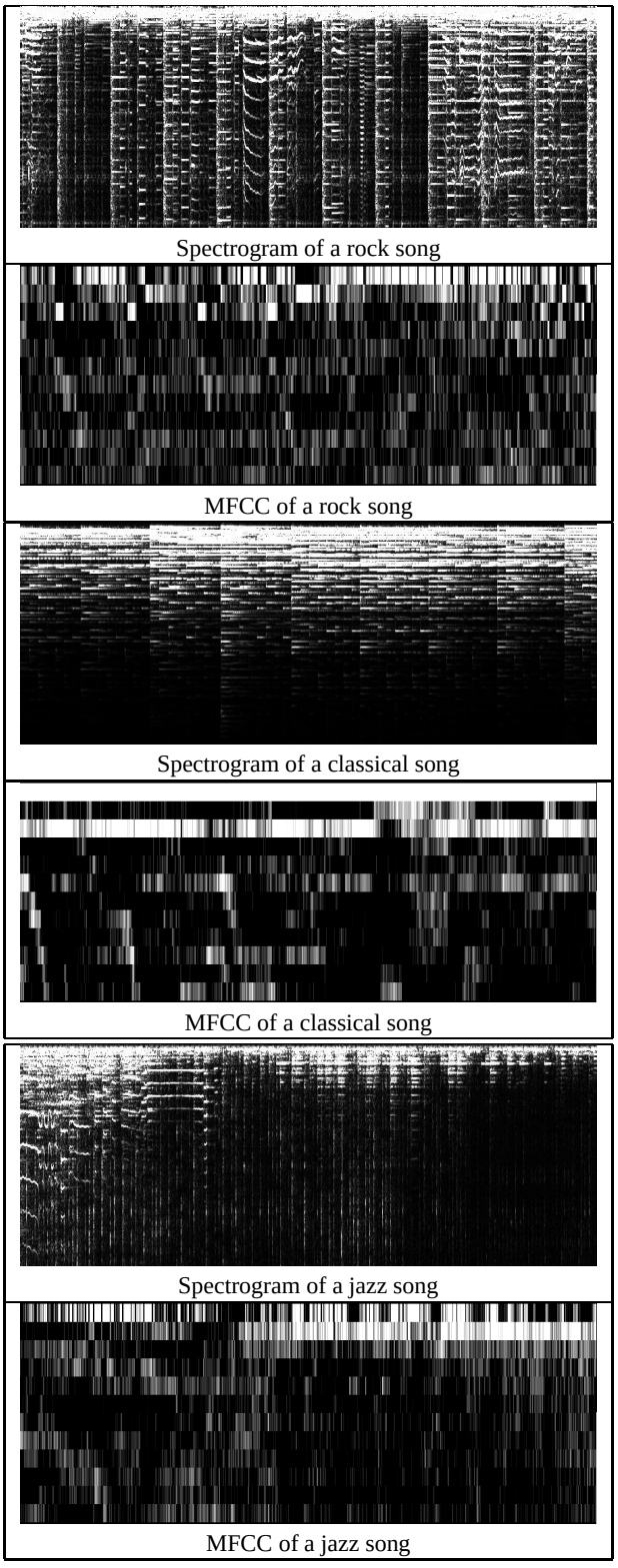

Table 1: *The above figures show images of spectrogram and MFCC data for rock, classic and jazz music.*

4. CLASSIFICATION

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.

4.1. 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.

4.2. Results

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.

4.3. 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.

5. 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, at least 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.

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.

6. CONCLUSION

In this paper 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 data points 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.

7. REFERENCES

[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.

[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.

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

[UN2001a] Nam, Unjung, Julius O. Smith III, and Jonathan Berger. (2001a) *Automatic Music Style Classification:towards the detection of perceptually similar music* In Proceedings of Japanese Society of Music Cognition and Perception, Fukuoka, Japan, May 2001.

[UN2001b] Nam, Unjung and Jonathan Berger. (2001b) *Addressing the same but different-different but similar problem in automatic music classification.* In Proceesings of International Symposium in Music Information Retrieval 2001, Bloomington, IN, October 2001.