# Music Classification

# Project Report

**CS429/529 Machine Learning**

**Team Member:**

**Chihyu Shen**

**Haijin He**

**Christina Xuan Yu**

**November 14, 2017**

# Theoretical Background For FFT and MFCC:

### FFT (fast Fourier transform):

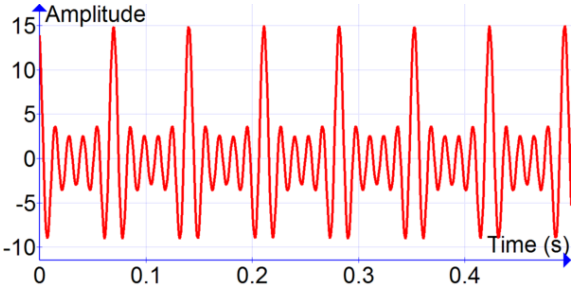
A fast Fourier transform (FFT) algorithm computes the discrete Fourier transform (DFT) of a sequence. In this project, it converts the signal from time domain to frequency domain. Comparing to DFT, it manages to reduce the complexity of computing the DFT from O(n^2) to O(n log n).

Discrete Fourier transform formula:

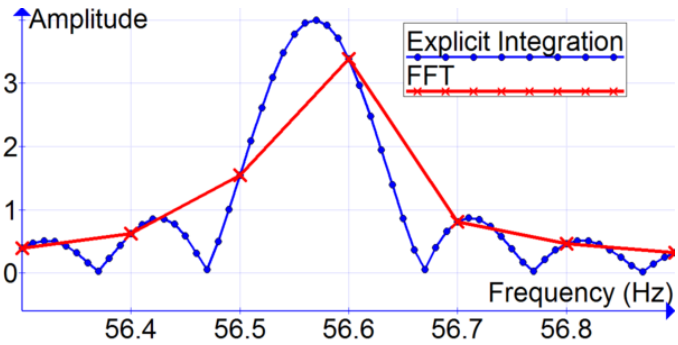
$$X_k = \sum_{n=0}^{N-1} x_n e^{-i 2\pi k \frac{n}{N}} \qquad k = 0, \ldots, N-1.$$

where N is size of signal

With a 5-level cosine signal in time domain:




The DFT and FFT results are:




Though the accuracy of FFT is not as precise as explicit integration is, the results are similar and FFT is 1000 times faster than the original computing of Fourier transform.

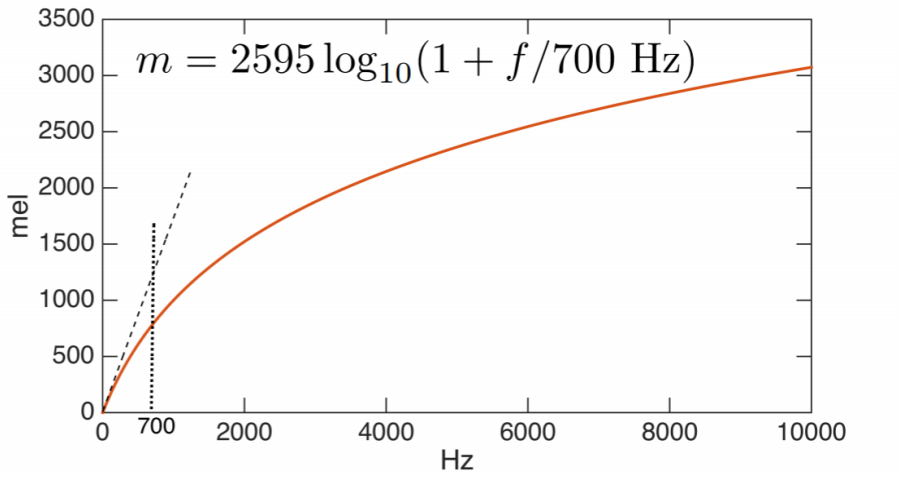
**MFCCs (Mel-Frequency Cepstrum Coefficients):**

Mel-Frequency Cepstrum is a non-linear frequency transform based on Mel scale, which is a good approach to mimic human auditory system. The process can be divided into the following parts:

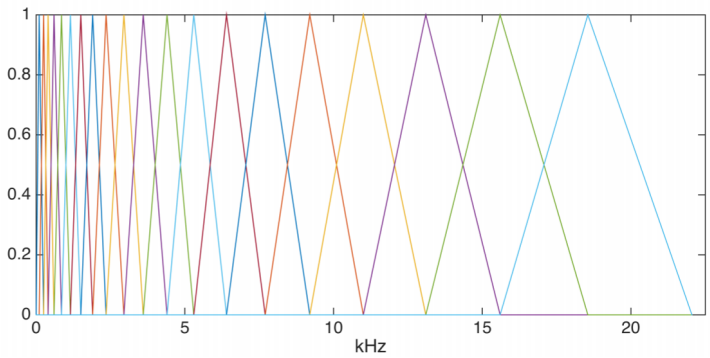
1. Divide the audio signal into signal frames.
2. Pre-ampplify the signal by passing it to a high-pass filter.
3. Apply Fourir transform to transform the signal to frequency domain.
4. Pass the spectrum of each signal frame through 20 Mel-scale triangular filters to obtain Mel-scaled signal.
5. Extract logarithm power from each Mel-scale.
6. Apply discrete inverse Fourier transform and transform to cepstrum domain.
7. MFCCs are the amplitudes of the cepstrum. In general, 12 coefficients plus power of signal frame are used to obtain a coefficient set of dimension of 13.

Frequency (Hz) to Mel-Scale (Non-linear transform):

| Hz | 40 | 161 | 200 | 404 | 693 | 867 | 1000 | 2022 | 3000 | 3393 | 4109 | 5526 | 6500 | 7743 | 12000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mel | 43 | 257 | 300 | 514 | 771 | 928 | 1000 | 1542 | 2000 | 2142 | 2314 | 2600 | 2771 | 2914 | 3228 |




Mel-scale triangular filters:



# 1. Describe your method for the third feature extraction

To started with designing the third method, we looked for some wavelet theory that is similar to the Fourier analysis. And we want to find something comprehensive and effective. We learnt Wavelet Transform of a function is the improved version of Fourier Transform, which captures both frequency *and* location information (location in time), and it is compact and fast.

Overfitting occurs when the model captures the noise of the data. Reducing the dimension of the feature space can help overcome the risk of overfitting. PCA (Principal Component Analysis) is a method for compressing a large dimensions of data into a set that captures the essence of the original data. It generates a new set of principal components by combining original variables linearly. All the principal components are orthogonal to each other so there is no redundant information in the new low-dimensional space.

Thus, we designed our third method for the feature extraction to be the following:

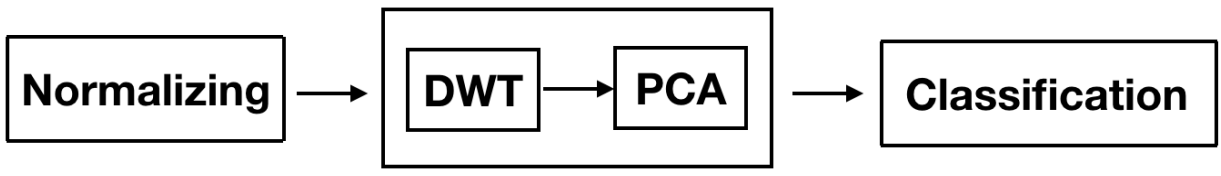
Firstly, we pre-process the data with normalization. Secondly, we transform the data with DWT to get the approximation coefficients:

*(cA, cD) = pywt.dwt(row, 'haar')*

Then we use PCA to select the features that we want to use:

*pca = PCA(n_components= numFeas)*
data = pca.fit_transform(data);

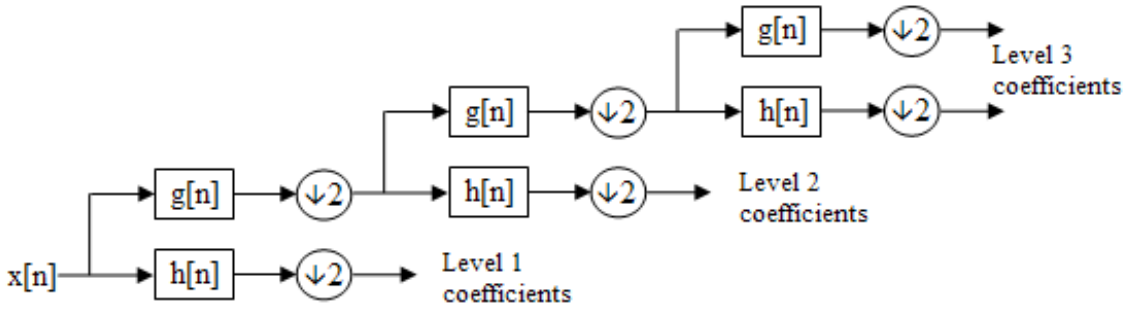
Lastly, we perform the classification using the preprocessed data.



## 2. Explain the rationale behind your feature design

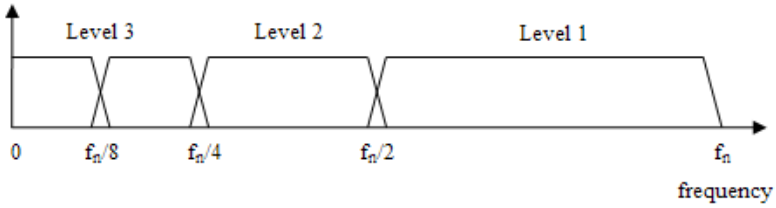
**DWT (Discrete Wavelet Transform):**

For this project, we implement one dimensional discrete wavelet transform(DWT). DWT was developed as an alternative to the short time Fourier Transform (STFT) to overcome problems related to its frequency and time resolution properties.
The rationale behind DWT involves low-pass filter, high-pass filter, downsampling, and Nyquist's rule.
The following is a 3-level structure for DWT.




x[n]: input audio signal
g[n]: low-pass filter, with half the cut-off frequency of the previous one
h[[n]: high-pass filter, with half the cut-off frequency of the previous one
↓2: downsample the signal by 2

After passing through low-pass filter and high-pass filter, half of the frequencies are removed. As a consequence, half of the signal samples can be discarded due to Nyquist's rule, and this is the reason to subsample the signal by 2. Moreover, downsampling the signal by 2 means to half the time resolution and double the frequency resolution. The following is the visualization of the frequency resolution regarding the 3 levels.




From the graph, one can tell that DWT provides high time resolution and low frequency resolution for high frequencies; and high frequency resolution and low time resolution for low frequencies.

**PCA (Principal Component Analysis):**

PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

Consider a data matrix X, the transformation is defined by a set of p-dimensional vectors of weights or loadings $w_{(k)} = (w1,...,wp)_{(k)}$ that map each row vector $x_{(i)}$ of X to a new vector of principal component scores $t_{(i)} = (t1,...,tm)_{(i)}$, given by $t_{k(i)} = x_{(i)}\, w_{(k)}$ ,for i=1,..,n, k=1,...,m
in such a way that the individual variables of t considered over the data set successively inherit the maximum possible variance from x, with each loading vector w constrained to be a unit vector.

For example, to calculate the first component:
In order to maximize variance, the first loading vector w(1) thus has to satisfy

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i (t_1)_{(i)}^2 \right\} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \sum_i \left( \mathbf{x}_{(i)} \cdot \mathbf{w} \right)^2 \right\}$$

Equivalently, writing this in matrix form gives

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \{ \|\mathbf{X}\mathbf{w}\|^2 \} = \arg\max_{\|\mathbf{w}\|=1} \left\{ \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \right\}$$

Since w(1) has been defined to be a unit vector, it equivalently also satisfies

$$\mathbf{w}_{(1)} = \arg\max \left\{ \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}$$

The quantity to be maximised can be recognised as a Rayleigh quotient. A standard result for a positive semidefinite matrix such as $X^TX$ is that the quotient's maximum possible value is the largest eigenvalue of the matrix, which occurs when w is the corresponding eigenvector.
With $w_{(1)}$ found, the first principal component of a data vector $x_{(i)}$ can then be given as a score $t1_{(i)} = x_{(i)} \cdot w_{(1)}$ in the transformed co-ordinates, or as the corresponding vector in the original variables, $\{x_{(i)} \cdot w_{(1)}\}\, w_{(1)}$.

## 3. Describe each of the two classifiers and why you used them

The two classifiers we used are: Gaussian Naive Bayes and Support Vector Machine.

1) Gaussian Naive Bayes is based on applying Bayes' theorem with independence assumptions between features. We assume that for each possible discrete value y_k of Y - 10 musical genres, the distribution of each continuous X_i is Gaussian, and is defined by a mean and standard deviation specific to x_i and y_k. Then we can use MLE - Maximum likelihood estimates for μ_ik:

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

And MVUE - Minimum variance unbiased estimator for $\sigma_{ik}^2$ :

$$\hat{\sigma}_{ik}^2 = \frac{1}{(\sum_j \delta(Y^j = y_k)) - 1} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k)$$

Besides treating the sample as continuous data, another important reason why we choose GNB as one of the classifier is, GNB is simple, so we reduce the risk of overfitting.

2) Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification analysis. An SVM model maximizes the margin (maximizes the distance between it and the nearest data point of each class) as a linear classifier. SVMs can also efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping the inputs into high-dimensional feature spaces.

K<x, z> = <Φ(x), Φ(z)>

The reasons why we chose SVMs as one of the classifier are: SVMs provides high accuracy; Once a boundary is established, most of the training data become redundant, which means that small changes to data cannot greatly affect the hyperplane and SVM; SVMs are naturally resistant to overfitting and work very well when identifying boundary regions.

## 4. Contrast accuracy for the 2 classifiers and 3 feature types, provide confusion matrices

With 10-fold cross validation, the combination gave us the best accuracy is MFCC + SVM.

The highest accuracy 70% is generated by GNB + FFT without using 10 fold cross validation.

**Confusion matrices:**

GNB+FFT:

| - | Blues | Classical | Country | Disco | Hiphop | Jazz | Metal | Pop | Reggae | Rock |
|---|---|---|---|---|---|---|---|---|---|---|
| Blues | 19 | 3 | 24 | 0 | 5 | 14 | 14 | 4 | 4 | 3 |
| Classical | 1 | 36 | 15 | 1 | 12 | 11 | 12 | 1 | 0 | 1 |
| Country | 7 | 0 | 53 | 0 | 0 | 4 | 20 | 3 | 0 | 3 |
| Disco | 3 | 5 | 20 | 1 | 3 | 4 | 48 | 2 | 2 | 2 |
| Hiphop | 7 | 9 | 5 | 4 | 20 | 5 | 16 | 7 | 13 | 4 |
| Jazz | 0 | 5 | 31 | 0 | 7 | 35 | 6 | 4 | 2 | 0 |
| Metal | 3 | 0 | 14 | 4 | 3 | 0 | 60 | 6 | 0 | 0 |
| Pop | 9 | 3 | 19 | 6 | 5 | 3 | 19 | 17 | 0 | 9 |
| Reggae | 7 | 4 | 20 | 2 | 7 | 9 | 10 | 2 | 23 | 6 |
| Rock | 4 | 1 | 18 | 1 | 2 | 6 | 40 | 3 | 13 | 2 |

GNB+MFCC:

| - | Blues | Classical | Country | Disco | Hiphop | Jazz | Metal | Pop | Reggae | Rock |
|---|---|---|---|---|---|---|---|---|---|---|
| Blues | 35 | 2 | 8 | 9 | 2 | 4 | 13 | 0 | 9 | 8 |
| Classical | 4 | 37 | 17 | 3 | 1 | 19 | 0 | 2 | 3 | 4 |
| Country | 12 | 3 | 30 | 12 | 0 | 2 | 2 | 6 | 20 | 3 |
| Disco | 4 | 1 | 3 | 42 | 1 | 3 | 5 | 15 | 9 | 7 |
| Hiphop | 14 | 0 | 3 | 13 | 7 | 6 | 2 | 9 | 33 | 3 |
| Jazz | 16 | 9 | 15 | 4 | 1 | 19 | 0 | 5 | 19 | 2 |
| Metal | 4 | 2 | 1 | 9 | 2 | 0 | 66 | 0 | 5 | 1 |
| Pop | 8 | 2 | 4 | 14 | 3 | 0 | 0 | 46 | 10 | 3 |
| Reggae | 6 | 0 | 12 | 5 | 2 | 4 | 3 | 4 | 51 | 3 |
| Rock | 12 | 3 | 14 | 24 | 2 | 0 | 5 | 2 | 15 | 13 |

GNB+DWT:

| - | Blues | Classical | Country | Disco | Hiphop | Jazz | Metal | Pop | Reggae | Rock |
|---|---|---|---|---|---|---|---|---|---|---|
| Blues | 0 | 60 | 10 | 14 | 0 | 0 | 6 | 0 | 0 | 0 |
| Classical | 1 | 71 | 8 | 10 | 0 | 0 | 0 | 0 | 0 | 0 |
| Country | 0 | 57 | 10 | 16 | 1 | 0 | 4 | 2 | 0 | 0 |
| Disco | 0 | 21 | 10 | 28 | 0 | 0 | 27 | 2 | 2 | 0 |
| Hiphop | 0 | 25 | 11 | 13 | 5 | 1 | 16 | 8 | 11 | 0 |
| Jazz | 1 | 52 | 7 | 6 | 0 | 4 | 19 | 0 | 0 | 1 |
| Metal | 0 | 7 | 5 | 23 | 1 | 0 | 54 | 0 | 0 | 0 |
| Pop | 0 | 22 | 17 | 18 | 6 | 0 | 17 | 9 | 1 | 0 |
| Reggae | 0 | 53 | 14 | 5 | 0 | 0 | 8 | 3 | 7 | 0 |
| Rock | 0 | 39 | 7 | 18 | 2 | 0 | 19 | 2 | 3 | 0 |

SVM+FFT:

| - | Blues | Classical | Country | Disco | Hiphop | Jazz | Metal | Pop | Reggae | Rock |
|---|---|---|---|---|---|---|---|---|---|---|
| Blues | 10 | 0 | 50 | 0 | 1 | 3 | 23 | 3 | 0 | 0 |
| Classical | 8 | 5 | 41 | 1 | 5 | 7 | 21 | 1 | 1 | 0 |
| Country | 0 | 0 | 54 | 0 | 0 | 0 | 36 | 0 | 0 | 0 |
| Disco | 1 | 0 | 21 | 0 | 2 | 1 | 64 | 1 | 0 | 0 |
| Hiphop | 2 | 0 | 48 | 0 | 2 | 2 | 34 | 1 | 1 | 0 |
| Jazz | 1 | 0 | 64 | 0 | 0 | 9 | 16 | 0 | 0 | 0 |
| Metal | 0 | 0 | 20 | 0 | 0 | 0 | 70 | 0 | 0 | 0 |
| Pop | 1 | 0 | 51 | 0 | 2 | 0 | 34 | 2 | 0 | 0 |
| Reggae | 0 | 0 | 54 | 1 | 1 | 11 | 16 | 5 | 1 | 1 |
| Rock | 1 | 0 | 33 | 0 | 0 | 4 | 49 | 1 | 0 | 2 |

SVM+MFC:

| - | Blues | Classical | Country | Disco | Hiphop | Jazz | Metal | Pop | Reggae | Rock |
|---|---|---|---|---|---|---|---|---|---|---|
| Blues | 31 | 5 | 10 | 4 | 6 | 5 | 9 | 1 | 8 | 11 |
| Classical | 3 | 65 | 7 | 2 | 0 | 10 | 0 | 1 | 1 | 1 |
| Country | 16 | 10 | 20 | 7 | 0 | 20 | 2 | 5 | 1 | 9 |
| Disco | 3 | 2 | 6 | 47 | 7 | 4 | 2 | 3 | 4 | 12 |
| Hiphop | 3 | 0 | 2 | 8 | 33 | 4 | 5 | 7 | 23 | 5 |
| Jazz | 5 | 19 | 14 | 5 | 7 | 30 | 0 | 3 | 4 | 3 |
| Metal | 8 | 0 | 0 | 4 | 9 | 0 | 65 | 0 | 1 | 3 |
| Pop | 0 | 3 | 6 | 9 | 6 | 2 | 0 | 62 | 1 | 1 |
| Reggae | 10 | 3 | 5 | 6 | 15 | 4 | 3 | 2 | 38 | 4 |
| Rock | 21 | 5 | 5 | 11 | 5 | 8 | 5 | 4 | 4 | 22 |

SVM+DWT:

| - | Blues | Classical | Country | Disco | Hiphop | Jazz | Metal | Pop | Reggae | Rock |
|---|---|---|---|---|---|---|---|---|---|---|
| Blues | 16 | 15 | 18 | 4 | 5 | 10 | 2 | 7 | 7 | 6 |
| Classical | 15 | 13 | 14 | 4 | 6 | 12 | 1 | 9 | 11 | 5 |
| Country | 25 | 9 | 7 | 4 | 12 | 9 | 3 | 9 | 8 | 4 |
| Disco | 17 | 15 | 12 | 4 | 11 | 9 | 1 | 7 | 7 | 7 |
| Hiphop | 13 | 15 | 18 | 3 | 13 | 10 | 0 | 4 | 7 | 7 |
| Jazz | 14 | 13 | 9 | 4 | 3 | 33 | 0 | 3 | 4 | 7 |
| Metal | 14 | 15 | 13 | 1 | 4 | 19 | 12 | 4 | 7 | 1 |
| Pop | 12 | 16 | 14 | 1 | 10 | 14 | 1 | 3 | 7 | 12 |
| Reggae | 18 | 15 | 12 | 5 | 4 | 11 | 1 | 7 | 10 | 7 |
| Rock | 23 | 8 | 22 | 4 | 10 | 8 | 1 | 2 | 7 | 5 |

## 5. Describe results and provide an explanation for bias

The results shows that generally MFCC performs better than other methods. It is probably MFCC captures higher level feature than FTT of DWT, which is kind raw feature of the music. We know that music genre classification is highly subjective, relying heavily on high level features like rhyme, pitch, timbre.
By looking at the confusion matrixes, we can see the bias of a classifier. It may be caused by the distribution model can not represent the data well, and underfitting happened.

## 6. Describe how could you improve further this classification task

The classification task can be improved by using  a more powerful classifier, such as neural network and deep learning technique. From this presentation, the author implemented deep learning on GTZAN dataset and achieved 84% accuracy.
We would also feed the classifier with more various training data for each music type.