Ukrainian Catholic University

Faculty of Applied Sciences

Computer Science & Business Analytics Programmes

Music Genre Classification

Linear Algebra final project report

Authors:
Khrystyna Kuts
Victor-Mykola Muryn
Yurii Sahaidak

May 15, 2024

Contents

Abstract

This project explores the classification of music genres using linear algebra tools, primarily Fast Fourier Transform (FFT) and Singular Value Decomposition (SVD). FFT converts audio signals into frequency-domain representations, and SVD is applied for dimensionality reduction and feature extraction. The classification component leverages Support Vector Machines (SVM), relying on principles of linear algebra for its operation. The model was trained on the GTZAN dataset. Using the presented methods, the accuracy percentage for the 10 music genres is 80.00%, and the F1 score is 80.02%.

Key words: Linear Algebra, Music Classification, Fourier Transform, Singular Value Decomposition, Support Vector Machine

1 Introduction

1.1 Research area

Music genre classification is a classical task in signal processing and is significant in various domains. It is used in personalized music recommendation systems, music streaming platforms, and content organization in digital libraries. By accurately categorizing music based on its genre, listeners can easily discover new songs or artists that align with their preferences, enhancing their overall music listening experience. Moreover, such classification methods enable efficient content management, targeted marketing, and audience segmentation for the music production and distribution industries.

1.2 Projects aim and tasks

Our objective is to delve into applying linear algebra techniques in signal classification. The project's tasks entail an in-depth exploration of the mechanics of Fast Fourier Transformation, Singular Value Decomposition, and Support Vector Machines. To gain a comprehensive understanding, we will implement SVD method. Then, we will use mentioned methods to classify music genres.

2 Literature review

2.1 Possible approaches

Each audio record initially is a vector of a very large size. For example, the representation of a 30-second audio clip in the frequency domain with in our case is a vector of size (1,660 000). Therefore, one should process the data to obtain a representation that is more suitable for further classification.

The paper "Classification of music genres using sparse representations in overcomplete dictionaries" -[1] describes one way to classify music genres using dictionary learning (DL). For data processing, the authors use the Linear Predictive Coding tool and the K-SVD algorithm, and for classification, they utilize Orthogonal Matching Pursuit (OMP), which is typical for dictionary learning (DL).

As the aim of our project is to delve into applying linear algebra techniques, for data processing we chose Mel-frequency cepstral coefficients (MFCCs), which include FFT, and Singular Value Decomposition (SVD) techniques, and for classification, we opted for Support Vector Machine (SVM). All of these methods use linear algebra.

2.2 Data processing

The first step is to transform our audio into a function with which we will work further. The perfect way to do this is to apply a Fourier Transformation, which describes the extent to which various frequencies are present in the original function. The Fourier Transformation works only with continuous functions, and the sound is discrete; we should apply Discrete Fourier Transformation (DFT). The drawback of DFT is its time complexity of $O(n^2)$. In 1965, James Cooley of IBM and John Tukey of Princeton published a paper, "An Algorithm for the Machine Calculation of Complex Fourier Series" [2], in which they described an algorithm that makes a Discrete Fourier Transformation in $O(nlogn)$ and called it Fast Fourier Transformation.

The second step of processing data will be a Singular Value Decomposition. This technique was invented by two independent mathematicians, Eugenio Beltrami and Camille Jordan, more than 100 years ago. SVD allows us to reduce the number of dimensions of the sample, which is an essential feature in further steps. This step allows us to train our model faster, requiring fewer computing resources.

An alternative to the SVD method is the Principal Component Analysis (PCA), which Karl Pearson invented in 1901 [3]. Though both techniques perform similar tasks, the SVD method has a wider range of applications, such as matrix approximation and recommendation systems. In our specific problem, either method could be used interchangeably. However, we opted for the SVD approach for its popularity and versatility.

2.3 Classification

The next step of the project is to make a classification itself. We want to use a Support Vector Machine (SVM, also Support Vector Networks) invented initially by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1964, then modernized in 1992 by Bernhard Boser, Isabelle Guyon and Vladimir Vapnik [4]. An alternative way to classify is to apply Neural Networks, but it is not relevant for this project.

3 Methodology

3.1 Pipeline of implementation

3.1.1 Pipeline overview

The analysis consists of several processes. First of all, the data is prepared: the samples are divided into pieces of 20 milliseconds, and those smaller samples are gathered into one array. The first stage of the analysis is FFT processing. For each element in the array, FFT is applied, and the results are stored as a second dimension of the array. Later, the results are optimized with the help of the Mel frequency spectrogram. The third step is the reduction of the samples via SVD. This will reduce the size of the second dimension by ignoring less significant frequencies. At the end, SVM will be applied to get a dataset classifier.

3.1.2 Mel-frequency

Mel-frequency cepstral coefficients (MFCCs) are feature representation in audio signal processing. This step will help to obtain the features on the basis of which the audio classification will be done later.

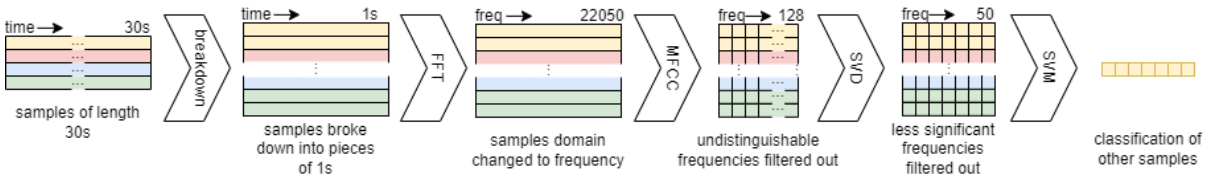

Figure 1: the implementation pipeline

The main idea behind MFCCs is that the human auditory system doesn't perceive all frequencies equally. Instead, it's more sensitive to changes in frequency at lower frequencies than at higher frequencies. The Mel scale is designed to mimic this perceptual characteristic.

The procedure for obtaining MFCCs consists of several steps:

a. Pre-emphasis: This step involves applying a pre-emphasis filter to the signal to amplify high frequencies, which helps in improving the signal-to-noise ratio.

$$y(n) = x(n) - \alpha \cdot x(n-1)$$

- $y(n)$ is the output signal
- $x(n)$ is the input signal
- α is the pre-emphasis coefficient

b. Framing: The pre-emphasized signal is divided into short frames, typically 20-40 milliseconds in duration, with overlap between successive frames. Each frame of the signal is multiplied by a window function (e.g., Hamming window) to reduce spectral leakage.

$$w(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right)$$

- n is the sample index ranging from 0 to $N-1$,
- N is the length of the window,
- $w(n)$ is Hemming window

c.Fourier Transform: For future signal analysis, it is transformed to the frequency domain. Fast Fourier Transformation(FFT), an optimized implementation of Discrete Fourier Transformation(DFT), is applied to do so.

The sequence of magnitude values in the time domain is transformed via the formula. The DFT matrix can be computed and applied to the signal based on the formula. Still, the computation complexity is $O(n^2)$.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{k}{N} n}$$

- $\{x_i\}$ - sample in the time domain
- $\{X_i\}$ - sample in the frequency domain
- N - number of elements in sample

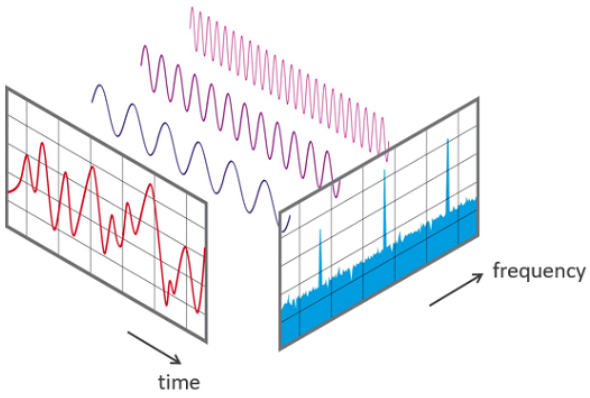

Figure 2: the FFT transformation result

- $k = 0, ..., N - 1$ - index of sample component

The FFT algorithm is used to improve the complexity. It decomposes the DFT matrix into a sequence of space matrices, allowing skip computations and decreasing the complexity to $O(log(n)$.

At this step, the domain of the sample is changed, but the dimension of the domain is not changed. According to Nyquist-Shannon theorem:

$$F_{max} = \frac{F_s}{2}$$

- $\{F_{max}\}$ - the largest determined frequency

- $\{F_s\}$ - sample rate

the largest frequency that can be determined is half of the sample rate. Thus, the dimension of FFT transformation will not change because it stores both positive and negative values. Further computations allow compression of the sample.

d.Mel Filtering: The power spectrum obtained from the FFT is passed through a bank of filters spaced according to the mel scale, which is a perceptual scale of pitches. These filters are designed to mimic the human auditory system's response to different frequencies and are determined by the number of frequencies that should be left after MFCC application.

e.Logarithmic compression: A human ear distinguishes relative differences between signals. Thus, the important frequencies should be logarithmically distributed. The logarithm function is applied to the filterbank energies to mimic this dependency.

After the MFCC application, each sample of 660,000-dimensional time domain space is transformed into 128-dimensional frequency domain space. The data matrix obtains a size of 48000x128.

3.1.3 Data reduction: SVD

After the MFCC step, the matrix of songs is of size 48000×128, which is too much since it has information that is not important for classification. The next step of processing is to reduce the redundant information.

The Singular Value Decomposition (SVD) algorithm is a powerful tool employed for dimensionality reduction in data processing tasks. Given a matrix $A_{m \times n}$, SVD factorizes it into three constituent matrices: $A = U\Sigma V^\top$. Here, the matrices are defined as follows:

- $U_{m\times m}$ represents a matrix of orthonormal eigenvectors of AA^T;
- $V^\top_{n\times n}$ denotes the transpose of the matrix containing the orthonormal eigenvectors of A^TA;
- $\Sigma_{m\times n} = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ is a diagonal matrix with k elements, each corresponding to the square root of the positive eigenvalues of either AA^T or A^TA (both matrices possess the same positive eigenvalues).

Each eigenvalue λ_i in the set $\Lambda := \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ reflects the significance of the associated eigenvector in the matrix $V^\top$. To discard non-significant information, the rank of the matrices can be reduced. For example, if it is observed that a rank $r < k$ encompasses 80% of the information while the remaining dimensions account for only 20% of the data, those latter dimensions can be omitted. Consequently, Σ is reduced to size $r \times r$, and the last $n - r$ rows of $V^\top$ and the last $m - r$ columns of U are discarded. The reduced SVD comprises matrices $U_{m\times r}, \Sigma_{r\times r}, V^\top_{r\times n}$.

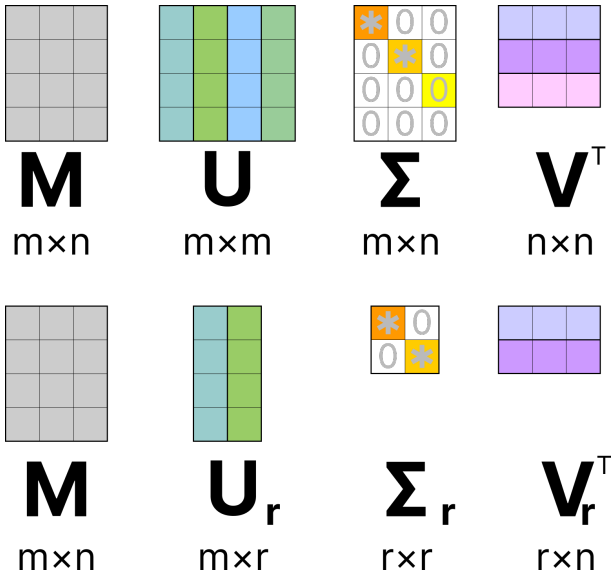

Figure 3: Resulted matrices of SVD [5]

The resultant matrix $U\Sigma V^\top$ retains 80% of the information while possessing a reduced rank of r.

Determining the appropriate rank r for reduction necessitates a specific criterion. Consider summing the eigenvalues up to k as $\sum_{n=1}^{k} \lambda_n$, which is taken to be 100%. By sorting all eigenvalues in descending order and selecting the first r values, the cumulative sum of these selected eigenvalues typically ranges from 60% to 80% of $\sum_{n=1}^{k} \lambda_n$.

The reduction of matrix A is achieved via SVD: $A = U\Sigma V^\top$. Subsequently, a reduced orthogonal basis $\Sigma_r V_r^\top$ is obtained. Each song M_n can then be expressed in terms of $\Sigma_r V_r^\top$ as $M_n = U_n \Sigma_r V_r^\top \implies U_n = M_n V_r \Sigma_r^{-1}$. Thus, the songs are now represented in the r-dimensional space instead of the original space derived from the Mel-frequency cepstrum.

Before the SVD, the matrix of songs was of shape 48000×128, but now 48000×50. The matrices U_n are now primed for classification.

3.1.4 Classification: SVM

Support Vector Machines (SVM) is a supervised learning algorithm used for classification tasks. SVM works by finding the optimal hyperplane that best separates the data points of different classes in a high-dimensional feature space. Decision function of the classifier is activation function g on equation of a hyperplane. g uses to transform results of $w^T x+b$ to the -1 or 1 according to the class labels.

$$y \in \{-1, 1\}$$

$$h(x) = g(w^T x + b)$$

where:

- y - class labels
- $h(x)$ - hypothesis or decision function
- x represents the input data point
- w is the weight vector perpendicular to the hyperplane
- b is the bias term

For example, in a two-dimensional space, a hyperplane is a line. In a three-dimensional space, it's a plane. And in higher dimensions, it's a hyperplane.

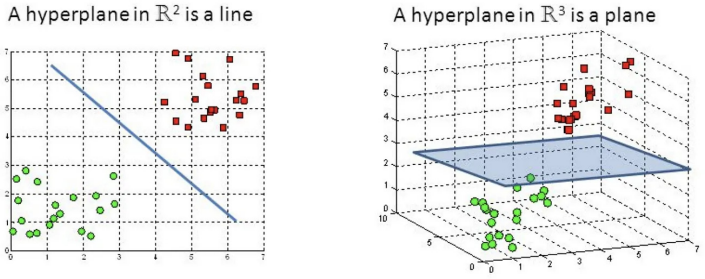

Figure 4: An image of hyperplanes in R^2 and R^3 [6]

The best hyperplane is the one that maximizes the margin - the distance between the hyperplane and the observations of each class closest to the hyperplane (support vectors). This condition form such optimization problem with following objective function:

$$\arg\min_{w,b} \frac{1}{2}||w||^2$$

Subject to the constraints:

$$y_i(w^T x_i + b) \geq 1 \text{ for } i = 1, 2, ..., n$$

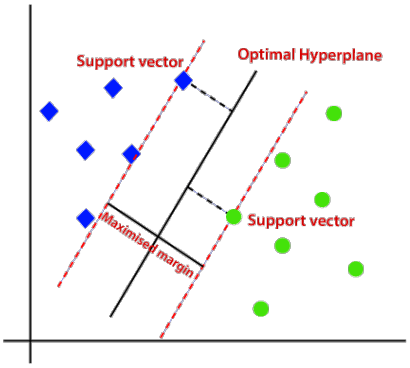

Figure 5: Illustration of the margins and support vectors [7]

This part $y_i(w^T x_i + b) \geq 1$ ensures that all classes are classified correctly. In case of non-linearly separable data there is objective function with introduced error term ξ:

$$\arg\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i$$

Subject to the constraints:

$$y_i(w^T x_i + b) \geq 1 - \xi_i \text{ for } \xi_i \geq 0, \quad i = 1, 2, ..., n$$

where :

- C - is the regularization parameter, controlling the trade-off between maximizing the margin and minimizing the classification error

- n - number of samples

Since we are working with musical genres that can overlap in practice, we assume that our data is not linearly separable.To classify non-linearly separable data, SVMs utilize the kernel trick. Typically, the strategy involves treating each sample in the training set as a landmark and then calculating the distance between each landmark and a training point using a kernel function. This procedure transforms the data into a higher-dimensional space where it becomes linearly separable.

Having such function of our hyperplane:

$$h(x, w) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$$

Using kernel trick we can replace with:

$$h(x, w) = w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4 + w_5 f_5$$

$$f_1 = x_1, f_2 = x_2, f_3 = x_1 x_2, f_4 = x_1^2, f_f = x_2^2$$

The kernel function can be any that satisfies Mercer's theorem: a function K is a kernel function if it is symmetric, continuous and positive semi-definite. In our case we used radial basis function (RBF).

3.2 Implementation testing

We used a "GTZAN Dataset - Music Genre Classification" [8] dataset to test implementation. This dataset contains 1000 songs. Each of the 10 genres has 100 songs, 30 seconds each. The dataset was divided into 80% and 20% for train and test data accordingly. To measure the effectiveness of our classificator, we will use some metrics like accuracy and precision.

Accuracy is a simple method of measuring performance of a classificator with a given formula.

$$\text{Accuracy} = \frac{\text{Correct classifications}}{\text{All classifications}}$$

This method is helpful since our dataset has equal samples for each class. The other helpful metric is precision, which provides insights into the reliability of the model's predictions for each class.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

The total precision is calculated as a weighted sum of each class. It is also an average since the number of elements in the test set is equal.

The recall measures the ability of the classifier to identify all instances of a particular genre correctly.

Also, the F1 score is a harmonic mean of precision and recall. A high F1 score indicates that the classifier has high precision and recall, meaning it makes accurate positive predictions and captures most instances of the positive class.

A very helpful metric will also be a confusion matrix; it shows a number of correctly and incorrectly classified genres for each genre individually.

4 Results

This research was done in a Google Collab [9]. The results can be seen there, as well as below with explanations.

We worked with several hyperparameters:

- n_{FFT} - number of frequencies returned by FFT

- n_{MFCC} - number of frequencies returned after MFCC compression

- n_{SVD} - dimension of the bases of SVD

The n_{FFT} is set to 22050 so as not to lose data on FFT processing. The reduction of this parameter makes no sense because FFT has the least context of the problem. The 128 value of n_{MFCC} was found by trial and error. It stores enough information for the SVD, but it still decreases the size of the matrix, which is essential for SVD computation. The value of n_{SVD} is set to 50. The further increment does not influence the accuracy of the machine that much but increases the computation time.

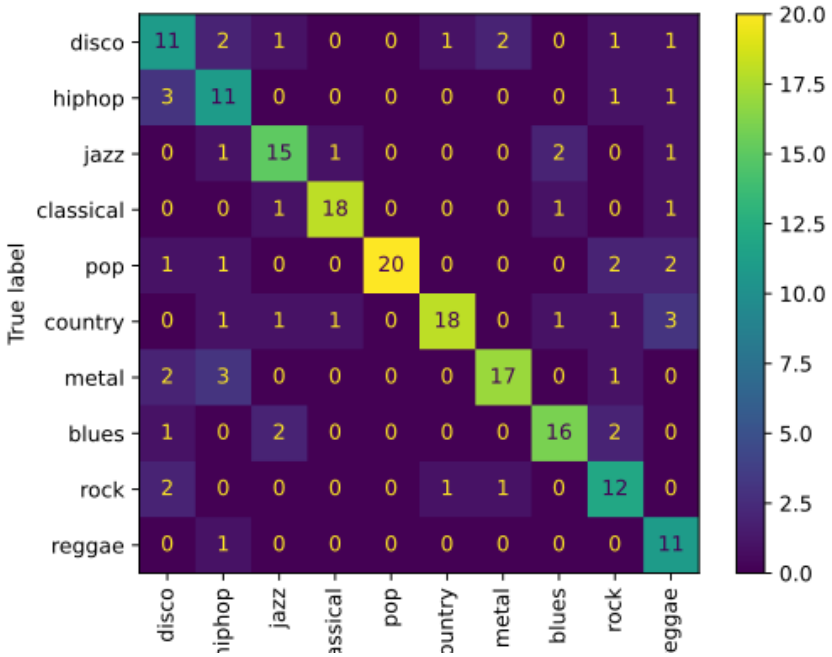

Figure 6: Confusion matrix of the trained model

The testing results showed that most of the songs were correctly classified. There are some inconsistencies in some classes. For instance, hip-hop was classified as disco three times, and the country was confused with reggae. This might be caused by the fact that it is hard to find songs that strictly belong to one genre and that some classes intersect with others.

The other results, such as f1 score, accuracy, recall, and precision, prove the model's accuracy. The obtained **accuracy** is 80.00%, **precision** - 80.45%, **recall** - 80.00%, **f1 score** value - 80.02%.

5 Conclusions

In the project, we explored the application of linear algebra methods. The FFT, MFCC, and SVD methods were investigated and applied to process audio samples. The SVM model was used as a classification model to implement a song genre classifier. The research gained experience of domain changing and domain dimension decreasing through the application of FFT and SVD on audio data. Moreover, work with MFCC provided a deep understanding of compression techniques based on human auditory perception. As a result of the work, the music genre classifier, which predicts song genres with a decent accuracy of 80%, was developed.

References

[1] Cristian Rusu. Classification of music genres using sparse representations in overcomplete dictionaries `https://eprints.imtlucca.it/1524/1/Classification%20of%20music%20genres%20using%20sparse.pdf`

[2] Cooley, J. W., & Tukey, J. W. (1965). An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematics of Computation, 19(90), 297–301. https://doi.org/10.2307/2003354

[3] Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. https://doi.org/10.1080/14786440109462720

[4] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (COLT '92). Association for Computing Machinery, New York, NY, USA, 144–152. https://doi.org/10.1145/130385.130401

[5] Mercurysheet, 2021, Visualization of Reduced SVD variants, https://upload.wikimedia.org/wikipedia/commons/c/c4/Reduced_Singular_Value_Decompositions.svg

[6] Medium, 2018, https://miro.medium.com/v2/resize:fit:1400/format:webp/1*ZpkLQf2FNfzfH4HXeMw4MQ.png

[7] Analitics Vidhya, 2024, https://editor.analyticsvidhya.com/uploads/729834.png

[8] GTZAN Dataset - Music Genre Classification. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

[9] Google Collab https://colab.research.google.com/drive/16K59uIY7wHkcv7fhsWIeWx5uyF3xjSE4