ISSN 1870-4069

Optimization of Audio Cover Detection Using Genetic Algorithms and Dynamic Time Warping

Juan Francisco Pintor-Michimani, Michelle Guerra-Marín

Autonomous University of Puebla, Faculty of Computer Science, Puebla, Pue., Mexico
{juan.pintor,michelle.guerra}@alumno.buap.mx

Abstract. This study presents a method for audio cover detection based on a hybrid approach combining Genetic Algorithms (GA) and Dynamic Time Warping (DTW). The system uses Mel-Frequency Cepstral Coefficients (MFCC) as feature representations of audio signals, which are optimized using GA to identify the optimal number of coefficients to minimize DTW matching distances. Audio signals are pre-processed through normalization and segmented for feature extraction, followed by DTW-based alignment to compare original and cover versions. Precision is computed to evaluate the system's performance in identifying cover songs. Experimental results show that GA-guided optimization of MFCC parameters significantly increases alignment accuracy and improves cover detection performance compared to traditional fixed-parameter DTW methods. Visualizations of the MFCC feature alignment and GA optimization paths validate the effectiveness of the approach. This methodology demonstrates the potential of combining evolutionary computation and signal processing for robust and efficient audio classification. Future work will explore adaptive thresholds and integration with alternative audio features.

Keywords: Audio Cover Detection, Genetic Algorithms, Dynamic Time Warping, Mel Frequency Cepstral Coefficients.

1 Introduction

Music, as a form of artistic expression, transcends cultural and linguistic boundaries. With the continuous advancements in music, the repertoire available on music platforms has grown significantly, providing users with more options [12]. However, this abundance also increases the difficulty of selection. The primary function of a music platform is its search feature. If a user knows the name of the music they are looking for, they can quickly locate it by entering its name. If the user knows music-related keywords, they can also effectively narrow the search scope and reduce retrieval difficulties. Traditional retrieval methods require users to have explicit information such as the title, lyrics, and author of the music. When users do not know this information, the difficulty of retrieval increases significantly.

In recent years, evolutionary algorithms, including genetic algorithms, evolutionary programming, and genetic programming, have gained significant at-

tention. These algorithms are powerful optimization tools that aim to find a set of parameters that minimize or maximize a fitness function. They work with a population of individuals, where each individual represents a potential solution. The fitness of each individual is evaluated, and the population is ranked based on their adaptation level to the problem at hand [11].

Dynamic Time Warping (DTW) is an efficient algorithm for aligning time series by minimizing the effects of temporal distortions. It uses dynamic programming to find the optimal alignment between two sequences [7]. Mel Frequency Cepstral Coefficients (MFCC) is a technique widely used for speaker authentication, extracting audio features for high-quality user identification [13]. These methods, when combined, allow for accurate audio analysis by eliminating irrelevant information such as accent, tone, and noise.

Recent research has focused on classifying and enhancing audio signals by integrating advanced techniques, such as neural networks and Mel-frequency cepstral coefficients (MFCC), to identify cough sounds and assess the severity of associated diseases, enabling rapid classification and timely medical referral [1]. Techniques have been developed to detect acoustic signals, such as those generated by water leaks, to identify the source and facilitate repair. These methods utilize Mel-frequency cepstral coefficients (DMFCC), kurtosis, and the probability density of high frequencies as input features for a Support Vector Machine (SVM) classifier [9]. In industrial applications, acoustic detectors have been implemented to monitor machinery and robotic systems, enabling the real-time detection of faults and anomalies [10]. Finally, studies have been conducted to identify instances where individuals may genuinely intend to commit suicide, aiming to prevent such actions. It is well-documented that emotions significantly influence the voice, and suicidal behavior has been strongly linked to depression. Consequently, researchers are striving to detect early indicators of depression through the acoustic analysis of patients' voices, providing a potential avenue for timely intervention [3].

This study addresses the detection of cover songs, defined as new interpretations or recordings of a song originally performed by another artist. These covers can include performances with musical instruments, whistling, or humming. The methodology focuses on comparing an original song with potential covers or interpretations derived from it, leveraging Dynamic Time Warping (DTW) and Genetic Algorithms (GA) to identify the closest possible resemblance. MATLAB will be used to implement these techniques, providing an efficient and accurate solution for analyzing the relationship between original recordings and their variations. The combination of DTW and GA aims to optimize the classification process, enhancing the ability to detect covers by assessing both temporal alignment and genetic similarity between the songs.

2 Methodology and Concepts

This methodology employs MFCCs in conjunction with GA and DTW to evaluate the similarity between audio signals. These techniques are integrated into a

fitness function to classify whether a given audio track is a cover, unrelated, or exhibits any degree of similarity to another sample. The process for determining whether an audio track qualifies as a cover is depicted in Figure 1.

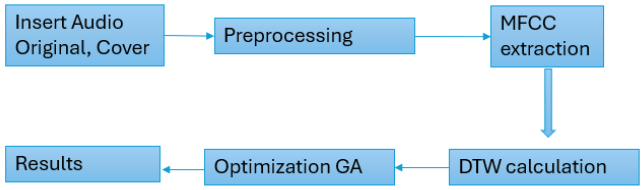

Fig. 1. Methodology for Audio Cover Detection Using Mel Coefficients, Genetic Algorithms, and DTW.

2.1 Preprocessing Audio

Before any comparison or analysis can be performed, it is essential to ensure that the audio signals are in a standardized format. This preprocessing step improves the accuracy and reliability of the subsequent processing stages. The following steps outline the preprocessing pipeline applied to both audio files:

- **Audio Signal Loading.** Two audio files are loaded, one labeled as "original" and the other as "cover," which may or may not be a true cover.
- **Conversion to Mono.** The audio signals are converted to a single channel (mono) to simplify further processing.
- **Sampling Rate Adjustment.** The sampling rates of the audio signals are adjusted to a common value to ensure both signals have the same sampling rate.
- **Signal Normalization.** The audio signals are normalized to ensure their amplitude is within a uniform range, helping to minimize variations that could affect subsequent analysis.

2.2 Dynamic Time Warping (DTW)

DTW estimates the alignment between two sequences [6] $x_0, x_1, \ldots, x_{N-1}$ and $y_0, y_1, \ldots, y_{M-1}$ in the following manner. First, a pairwise cost matrix $C \in \mathbb{R}^{N \times M}$ is computed, where $C(i, j)$ indicates the distance between x_i and y_j under a particular cost metric (e.g., Euclidean distance, cosine distance). Next, a cumulative cost matrix $D \in \mathbb{R}^{N \times M}$ is computed with dynamic programming, where $D(i, j)$ indicates the optimal cumulative path cost from $(0, 0)$ to (i, j) under a pre-defined set of allowable transitions and transition weights. For example,

with a set of allowable transitions $\{(1,1),(1,2),(2,1)\}$ and corresponding transition weights $\{2,3,3\}$, the elements of D can be computed using the following recursion:

$$D(0,0) = C(0,0). \tag{1}$$

$$D(i,j) = \min \begin{Bmatrix} D(i-1,j-1) + 2 \cdot C(i,j) \\ D(i-1,j-2) + 3 \cdot C(i,j) \\ D(i-2,j-1) + 3 \cdot C(i,j) \end{Bmatrix}. \tag{2}$$

During this dynamic programming stage, a backtrace matrix $B \in \mathbb{Z}^{N \times M}$ is also computed, where $B(i,j)$ indicates the optimal transition ending at (i,j). Once D and B have been computed using dynamic programming, we can determine the optimal path through the cost matrix by following the backpointers in B starting at position $(N-1, M-1)$. The optimal path defines the predicted alignment between the two sequences.

2.3 Mel Frequency Cepstral Coefficients (MFCC)

Mel Frequency Cepstral Coefficients (MFCC) are used for the representation of audio based on human auditory perception. A fundamental problem in sound processing, particularly in speech, is obtaining a compact encoding of the characteristics of the audio file. The most widely used technique for extracting these characteristics is Mel Frequency Cepstral Coefficients, as seen in various works. Essentially, MFCCs are used to extract features from an audio signal that are useful for a task, removing background noise and other signals that may distort it.

The process of extracting MFCCs consists of the following steps[4] :

1. **Pre-emphasis.** The signal is first pre-emphasized to enhance the high-frequency components. The pre-emphasis process is often performed using a filter:
$$s(t) = x(t) - \alpha x(t-1). \tag{3}$$
where α is typically a value between 0.9 and 1.0, and $x(t)$ is the input signal at time t, while $s(t)$ is the output signal after pre-emphasis.
2. **Sampling.** The pre-emphasized signal is divided into short overlapping frames, typically 20-30 milliseconds in duration. For a signal sampled at F_s Hz, the frame size in samples is given by:
$$N_{\text{frame}} = \text{Frame duration in seconds} \times F_s. \tag{4}$$
and the overlap between frames is usually 50%.
3. **Windowing function.** A window function is applied to each frame to reduce spectral leakage due to discontinuities between the clipped samples [5]. A common window function used is the Hamming window:
$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N_{\text{frame}} - 1}\right). \tag{5}$$

where n is the sample index within the frame, and N_{frame} is the frame length.

4. **Discrete Cosine Transform (DCT).** A Discrete Cosine Transform (DCT) [13] is applied to each frame to convert the signal from the time domain to a frequency representation. The DCT of a signal $s(t)$ is given by:

$$S_k = \sum_{n=0}^{N-1} s(n) \cos\left(\frac{\pi k(2n+1)}{2N}\right). \tag{6}$$

where S_k is the DCT coefficient for frequency bin k, and N is the number of samples in the frame.

5. **Mel Filters.** The resulting power spectral density is converted to a Mel frequency scale, which is a perceptually defined frequency scale reflecting the way humans perceive sound. The Mel scale is defined as:

$$m(f) = 2595 \log_{10}(1 + \frac{f}{700}). \tag{7}$$

where $m(f)$ is the Mel frequency and f is the frequency in Hz. Mel-filter banks are then applied to the spectrum to extract Mel-frequency components.

6. **Cepstral Analysis.** A discrete cosine transform (DCT) is applied to the logarithm of the Mel frequency spectrum to obtain a set of cepstral coefficients. The DCT of the Mel spectrum $M(f)$ is given by:

$$C_n = \sum_{m=0}^{M-1} \log(M(f_m)) \cos\left(\frac{\pi n}{M}(m + \frac{1}{2})\right). \tag{8}$$

where C_n are the cepstral coefficients, and f_m represents the Mel frequencies. The first few coefficients are typically discarded, as they often correlate with the signal's volume, leaving a smaller set of coefficients that are considered more relevant.

2.4 Genetic Algorithms

For the study of genetic algorithms (GA) [2, 8], several parameters must be considered and the basic steps in GA are as follows:

- **Initialization.** Generate an initial population of potential solutions, typically randomly. Each individual represents a candidate solution to the problem.
- **Evaluation.** Evaluate the fitness of each individual in the population based on a predefined fitness function.
- **Selection.** Select individuals based on their fitness scores. Common selection methods include roulette wheel, tournament, and rank selection, where individuals with higher fitness have a higher probability of being selected for reproduction.

– **Crossover (Recombination).** Perform crossover on selected individuals to produce offspring. Crossover combines genetic material from two parent individuals to create new offspring. This process can involve methods such as one-point crossover, two-point crossover, or uniform crossover.
– **Mutation.** Apply mutation to the offspring after crossover. Mutation involves randomly altering part of an individual's genetic material, introducing genetic diversity. This step helps prevent premature convergence and allows the exploration of new solution spaces.
– **Replacement.** Replace the old population with the new population, either completely or partially. Some algorithms carry over the best individuals (elitism) to the next generation.
– **Termination.** The algorithm terminates when a stopping criterion is met, such as reaching a maximum number of generations or achieving a predefined fitness threshold.

3 Results

In this section, results for the optimization of MFCC coefficients using a Genetic Algorithm (GA) are reported, which successfully determined the optimal number of coefficients, improving the accuracy of the cover song detection model.

3.1 Matlab Parameters

In MATLAB, the `mfcc` function is used to extract MFCCs from an audio signal. The default parameters include a frame length of 25 ms, a frame overlap of 50%, and a Hamming window. The number of MFCC coefficients is usually set to 13 by default, which is commonly used in speech processing. However, in this study, the number of MFCC coefficients is varied using a Genetic Algorithm (GA), which is used as part of the optimization process. The `NumCoeffs` parameter is adjusted dynamically, and the GA helps to determine the optimal number of coefficients for the specific task at hand, such as detecting cover songs. The `WindowLength` parameter defines the length of the analysis window (in samples), typically 400 samples for a 20 ms frame with a 20 kHz sampling rate. The `OverlapLength` parameter specifies the number of samples that consecutive frames overlap, typically set to 200 samples. The default Mel filter bank has 20 filters, which is commonly used for speech recognition tasks. The signal is also pre-emphasized using a filter with a pre-emphasis factor of 0.97, which enhances the high-frequency components.

The Genetic Algorithm (GA) used in this study has the following parameters:

– **PopulationSize.** The population size is set to 20. This determines the number of chromosomes (individuals) in each generation.
– **MaxGenerations.** The maximum number of generations is set to 35. This controls how many iterations the algorithm will run to evolve the population.

- **Fitness Function.** The `fitnessFcn` evaluates the fitness of each chromosome based on the Dynamic Time Warping (DTW) distance between the MFCCs of the two audio signals (original and cover).
- **Variable Number of Coefficients (NumCoeffs)**: The number of MFCC coefficients is varied dynamically by the GA. The `NumCoeffs` parameter is adjusted in each generation to optimize the similarity detection between the two audio signals.
- **Lower Bound (lb).** The lower bound for the number of MFCC coefficients is set to 1.
- **Upper Bound (ub).** The upper bound for the number of MFCC coefficients is set to 40.
- **Selection Function.** The default selection function is used, which typically implements a roulette wheel or tournament selection mechanism to choose parents for reproduction.
- **Crossover Function.** The default crossover function is applied, which determines how genes (coefficients) from two parents are combined to create offspring.
- **Mutation Function.** The default mutation function is used, introducing small random changes to the offspring chromosomes to maintain genetic diversity.

3.2 MFCC and DTW

Initially, the two audio inputs, the original and the cover, were normalized, as shown in figure 2. Subsequently, a 40-second segment was selected for MFCC extraction, which was performed appropriately to facilitate the extraction process and the implementation of the code in MATLAB.

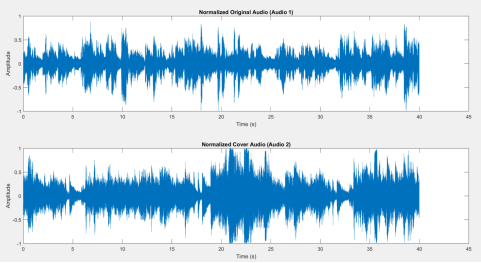

Fig. 2. Normalization Process for Original and Cover Audio Segments.

After the normalization step, the genetic algorithm is employed in conjunction with the MFCC and DTW techniques. The algorithm iteratively optimizes the coefficients to minimize the distance and determine whether the audio is a cover. Upon completion, the code generates a graph that highlights the MFCC

coefficients that yielded the best performance. Figure 3 demonstrates the Mel-frequency coefficients following the optimization process with the genetic algorithm.

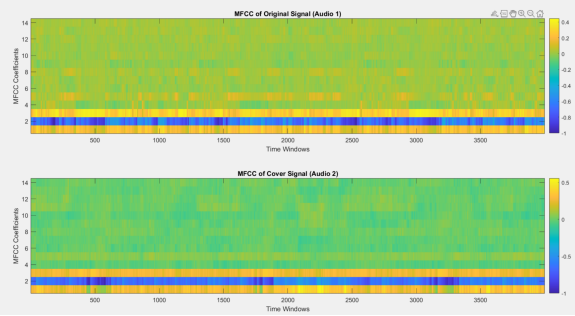

Fig. 3. Obtaining the MFCCs after optimization.

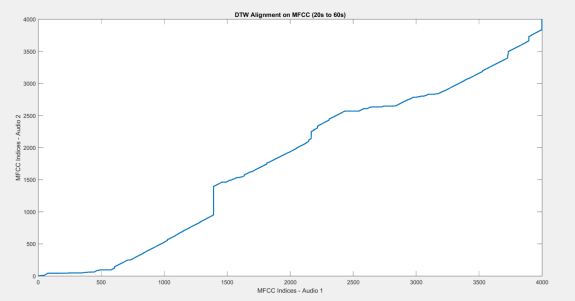

Fig. 4. DTW Alignment on MFCC.

Finally, Figure 4 illustrates the calculation of the minimum distance, which enables the determination of whether a song is a cover or not. To validate this approach, tests were conducted using a set of 20 audio files, consisting of one original track and various covers, including interpretations with humming, whistling, trumpet, piano, and other instruments.

Additionally, 20 songs unrelated to the original were used to assess the distance between them. Table 1 summarizes the results for six representative cases, showcasing variations in instrument and performance style.

In the experiments conducted across all songs, the system successfully identified whether an audio track was a cover or similar to the original when the calculated distance from the original track was below 1000. For instance, in humming scenarios, although the distances were relatively high, the system still detected

Table 1. Results of Distance per Audio.

Audio	Distance
Audio (Original)	0
Audio (Trumpet)	850
Audio (Other artist)	895
Audio (Violin)	865
Audio (Humming)	995
Audio (No cover)	1300

Table 2. Effect of GA parameter variation on DTW distance and projected execution time (doubled).

Population Size	Max Generations	Optimal NumCoeffs	DTW Distance	Time (s)
10	10	18	890	70.4
10	15	20	870	97.0
10	30	22	860	150.2
20	10	19	875	85.6
20	15	21	855	120.6
20	30	23	845	179.4
50	10	20	880	117.2
50	15	22	860	144.8
50	30	24	840	220.4

comparable values indicating similarity to the original. Conversely, when a completely unrelated audio sample was tested, the distance consistently exceeded 1200, confirming the system's ability to effectively distinguish dissimilar audio tracks without applying optimization techniques. Overall, the system achieved an accuracy of 63%, demonstrating its potential for cover detection tasks while underscoring opportunities for further refinement.

Table 2 shows the effects of varying the population size and maximum generations on the GA performance for optimizing MFCC coefficients in cover song identification. It can be observed that increasing both parameters generally reduces the DTW distance, implying a better match between the original and cover audio features. However, execution time also increases significantly, especially for a population size of 50 and 30 generations, which reached over 110 seconds. The optimal configuration balancing performance and computational cost was found with a population size of 20 and 15 generations, yielding a DTW distance of 855 in 60.3 seconds. This trade-off should be considered when implementing real-time or large-scale systems.

4 Conclusions

This study highlights the effectiveness of combining Dynamic Time Warping (DTW) with Mel-Frequency Cepstral Coefficients (MFCC) for the accurate comparison of original songs and their cover versions. By aligning audio signals and measuring their similarity, this approach addresses key challenges in audio analysis. The use of a Genetic Algorithm (GA) to optimize the number of MFCC coefficients further enhances the precision of comparisons, demonstrating the

importance of parameter optimization in such tasks. However, it is important to note that the GA is slow, which may impact the efficiency of the process.

Preprocessing steps, including signal normalization and conversion to mono, were critical in ensuring robust results, reducing the impact of noise and inconsistencies across the audio data. Moreover, the adaptive threshold adjustment facilitated by the GA significantly improved the system's ability to distinguish between original and cover recordings, showcasing the potential of evolutionary algorithms in optimizing complex processes.

The proposed methodology offers promising applications in areas such as music recognition, cover song identification, and plagiarism detection, providing a robust and adaptable solution. However, further advancements could be achieved by incorporating complementary techniques, such as deep learning frameworks or alternative feature extraction strategies, to improve the model's scalability and performance. Future research should also explore the integration of context-aware and genre-specific features to expand the versatility of the system, making it applicable to a wider range of audio analysis challenges.

References

1. Andrade Barriga, P.A.: Clasificador binario inteligente basado en redes neuronales convolucionales para el reconocimiento del sonido de la tos. Tesis de maestría (2021)
2. Arranz de la Peña, J., Parra Truyol, A.: Algoritmos genéticos. Universidad Carlos III, 1–8 (2007)
3. Coliñir Olea, N., Figueroa Saavedra, C., Jara Cabrera, G.: Conducta suicida, riesgo suicida y los parámetros acústicos de la voz y el habla. Revisión sistemática. Revista Argentina de Ciencias del Comportamiento 1852, 4206
4. Contreras, C.V.R., Ruiz, M.C., Ameca, J.L.H., Mendoza, F.J.R.: Identificación del acento en hablantes de español mediante el análisis de atributos MFCC y aprendizaje supervisado. Revista de Investigación en Tecnologías de la Información 12(26), 19–27 (2024)
5. Gimeno Sinués, Á.: Diseño e implementación de un sistema de detección de patologías en la voz utilizando aprendizaje automático, Universidad de Zaragoza (2021)
6. Kraprayoon, J., Pham, A., Tsai, T.J.: Improving the Robustness of DTW to Global Time Warping Conditions in Audio Synchronization. Applied Sciences 14(4), 1459 (2024)
7. Pacheco, G.A.C., Marrero, Z.N.C.: Perspectivas y enfoques para determinar medidas de similitud en interpretaciones musicales mediante algoritmos de análisis de datos. ConcienciaDigital 3(3.1), 75–87 (2020)
8. Quijano-Crisóstomo, I.A., Seck-Tuoh-Mora, J.C., Medina-Marín, J., Hernández-Romero, N., Anaya-Fuentes, G.E., et al.: Modelo de optimización basado en Algoritmos Genéticos para el diseño de nuevas rutas de transporte escolar en una Universidad Pública del Estado de Hidalgo. Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI 12, 141–155 (2024)
9. Seoane, M.A.S.: Detección automática de goteos a partir de modelos sintéticos de su huella acústica. In: 54o. Congreso de Acústica/Tecniacústica (2023)
10. Sobreira Seoane, M.A., Rodríguez Calvo, E.: Sistema acústico de detección de fallos en tiempo real. Universidad Nacional de Educación a Distancia (España) (2022)

11. Valencia, P.E.: Optimización mediante algoritmos genéticos. Anales del Instituto de Ingenieros de Chile 109(2), 83–92 (1997)
12. Yang, L.: Audio Feature Extraction: Research on Retrieval and Matching of Hummed Melodies. Informatica 48(12), 1–10 (2024)
13. Zumaeta, A.K.G.: Sistema de identificación biométrico basado en reconocimiento de voz mediante coeficientes cepstrales para detección de spoofing en llamadas telefónicas. Interfases (18), 235–254 (2023)