Detection of Similar Melodies by Repurposing Algorithms for Sequence Alignment and String Searching

Vihaan Krishnakumar[1] and Peter Washington[#]

[1]Archbishop Mitty High School, USA
[#]Advisor

ABSTRACT

Music plagiarism is an important concern for the music industry. Current methods of using experts to detect plagiarism are subjective and error-prone. This paper compares the performance of both string-searching algorithms and algorithms traditionally used in bioinformatics, and in particular, Knuth-Morris-Pratt (KMP) and Smith-Waterman, for the detection of melodic plagiarism. The input MIDI files are converted into an array after data processing and used as the basis for comparison. Across most thresholds, melodic plagiarism detection using KMP exhibits greater recall than, similar precision to, and faster runtimes than Smith-Waterman. We conclude that exact string searching algorithms like KMP can be more effective than local sequence alignment methods like Smith-Waterman.

Introduction

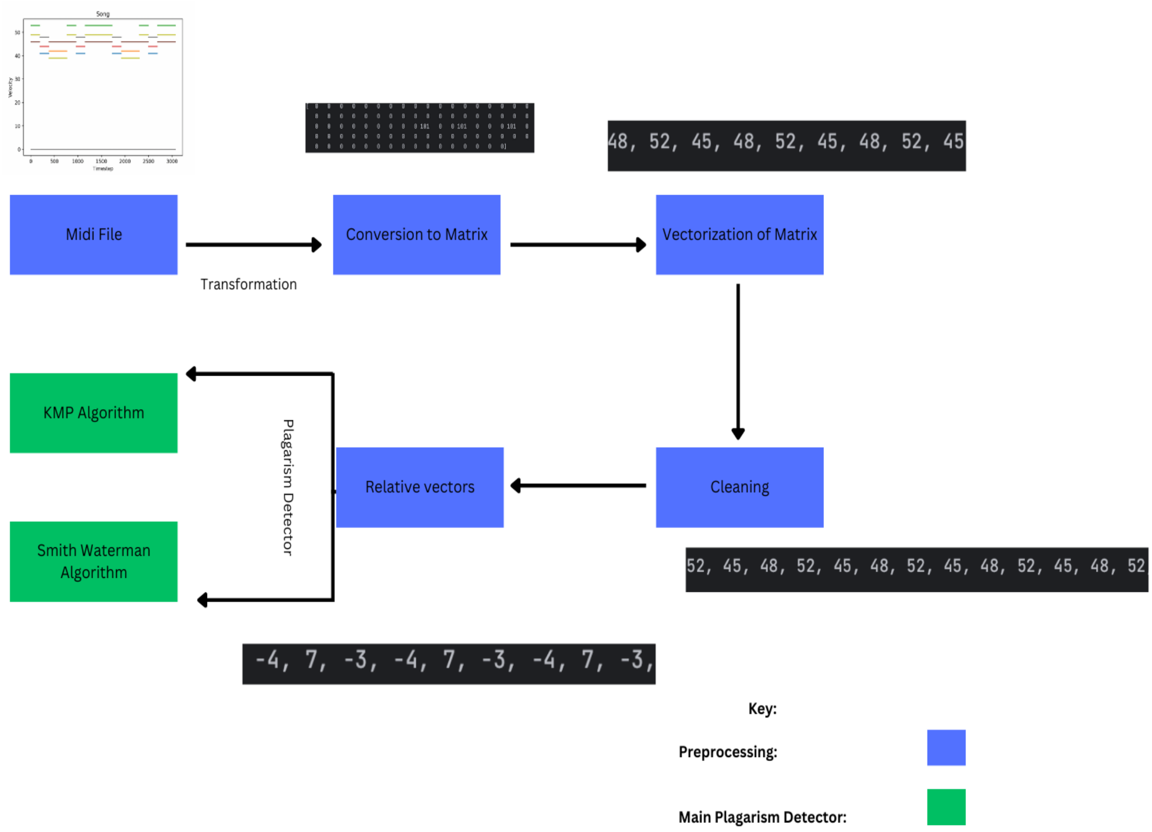
In today's music landscape, the issue of music plagiarism has caused significant financial losses, leading to multi-million-dollar lawsuits and potential job instability within the music industry [1]. Often, music producers unintentionally create compositions that bear striking resemblances to existing melodies, resulting in allegations of plagiarism. Recent high-profile cases, such as Ed Sheeran's legal battle over his song 'Thinking Out Loud,' alleged to have similarities with Marvin Gaye's 'Let’s Get it On' [2], shed light on the complexity of proving or disproving musical plagiarism, despite Sheeran's eventual legal victory.

This research paper delves into the existing methods and techniques for detecting similarities in musical compositions, aiming to address the challenges of identifying potential plagiarism. Previous studies have explored diverse approaches, such as converting MIDI files into grayscale images for analysis using Siamese CNN [3], employing path exploration over a binary mask [4], utilizing bipartite graph matching [5], and applying text-similarity and clustering algorithms [6].

The primary focus of this paper is to propose and discuss two main models—Smith-Waterman and Knuth-Morris-Pratt (KMP)—adapted for detecting similar melodies within songs. The adaptation of the Smith-Waterman model, known for aligning local DNA sequences, offers a potential solution for identifying resemblances in musical compositions. Similarly, the KMP model, originally used for pattern searching in texts, could be repurposed to seek out analogous melodies in music compositions. Notably, both models provide reliable and efficient similarity measurements crucial for effective detection.

To enable effective analysis, the research also introduces a method to convert MIDI files into a manageable data format. MIDI, a standard file type in music, contains information about notes, tempo, and velocity. Transforming this data into a matrix format—each row representing a time step and each column a note on the piano—facilitates the identification of played notes for analysis. Understanding that plagiarism in music extends beyond exact note replication and may involve variations in pitches, the paper proposes considering the distances between notes played to account for pitch changes. Using vector representation and relative arrays of the MIDI, the research aims to refine the data for application in the KMP and Smith-Waterman algorithms. In essence, the study endeavors to contribute to the ongoing efforts to detect music plagiarism and aims to offer insights into developing more robust and accurate methods for analyzing musical compositions.

Methods



```
[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0 101   0   0 101   0   0   0 101   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
```

```
48, 52, 45, 48, 52, 45, 48, 52, 45
```

```
52, 45, 48, 52, 45, 48, 52, 45, 48, 52, 45, 48, 52
```

```
-4, 7, -3, -4, 7, -3, -4, 7, -3,
```

Figure 1. The process of the Music Plagiarism Detector. The MIDI File is transformed into a matrix, converted into a vector, cleaned, and converted into relative arrays. The final result is fed into the KMP and Smith-Waterman Algorithms.

Pre-Processing

In the realm of musical data analysis, preprocessing steps play a crucial role in transforming raw MIDI files into formats conducive to computational analysis. This section explores the sequential steps involved in preparing MIDI data for further analysis.

I. Conversion of MIDI to Matrix

Raw MIDI files, a digital representation of musical notes, are transformed into matrices. These matrices encapsulate the velocities of each note (ranging from 0 to 127) at various time steps. This transformation allows a structured representation of the musical data, enabling subsequent computational processing.

II. Vectorization of Matrices

Matrices are further converted into vectors to facilitate enhanced readability and ease of computational analysis. Notes played are organized and appended into a list according to their respective time steps, simplifying subsequent data handling and analysis.

III. Removal of Repetitive Notes

To streamline computational processing and retain the melodic integrity of the musical piece, a function is implemented to eliminate repetitive notes within the list. This removal of redundant notes contributes to faster processing without compromising the musical essence.

IV. Relative Arrays

Another essential aspect of preprocessing involves the creation of relative arrays. These arrays are generated to analyze the relationships between note lengths by computing the edit distance between each note. This step aims to provide insights into the duration relationships within the musical sequence.

Each of these preprocessing methods is designed to make the musical data more manageable for computational analysis while ensuring that the fundamental musical characteristics are preserved.

The comprehensive preprocessing steps detailed in this section set the foundation for further computational analysis of MIDI data and compare performance across different algorithms.

KMP Function*:*

This function takes in both data that will be compared with each other, and a threshold that represents the minimum amount to be considered plagiarized. KMP Function or the Knuth Morris Pratt algorithm is a well-known algorithm that checks the occurrence of a word within another word. This function can be manipulated into taking a list of integers and with the help of a loop, it can check if two MIDI data are plagiarized. It will take the smaller list, and make sublists with the length, according to the threshold. These sublists will be compared with the bigger list, as both lists are relative lists of the MIDI data, the common pattern, or the plagiarized melodies can be found. The time complexity of this is O(n+m) [7].

Smith-Waterman Algorithm

The Smith–Waterman algorithm is a popular method to identify similarities in DNA. The algorithm performs local sequence alignment; that is, for determining similar regions between two strings of nucleic acid sequences or protein sequences. Instead of looking at the entire sequence, the Smith–Waterman algorithm compares segments of all possible lengths and optimizes the similarity measure. The Smith-Waterman Algorithm can be manipulated to take in a list of integers (the final result from the preprocessing functions) and align them according to the best alignment. This also gives which parts of the data are plagiarized.

Dataset

50 data points with plagiarized and unplagiarized melodies were used, which contained simple MIDI data, and the plagiarized ones were made with some similar data of varying degrees. These pieces of data were handmade by using MIDI files from Cymatics [8] and Unison [9,10] and manipulating them to make plagiarized and unplagiarized datasets. We originally selected simple melody files from different sources. Plagiarized data were created by infusing parts of the melody file into another melody file of varying degrees. For example, a small section of the original file was copied and pasted into another melody file at different spots. This creates a plagiarized file. Similarly, differing sizes of sections were copied and pasted on a second file to create multiple plagiarized sets.

Results

The KMP exhibited an average precision score of 0.972 and an average recall score of 0.88 with an average runtime of 416.54 seconds across the 6 thresholds that we evaluated. The Smith-Waterman algorithm exhibited an average precision score of 0.983 and an average recall score of 0.767 with an average runtime of 2025.57 seconds across the 6 thresholds we evaluated. KMP was more efficient and resulted in higher performance than Smith-Waterman. However, Smith-Waterman provided an opportunity to handle plagiarism despite gaps in the sequence, a functionality that is missing in the KMP algorithm.

Table 1. A table with the threshold, precision score, recall score, and time in seconds of both KMP and Smith-Waterman.

KMP Threshold	Precision	Recall	Total Time (s)
0.10	0.89	0.94	445.88
0.20	1.00	0.94	456.96
0.25	1.00	0.94	337.78
0.40	1.00	0.71	425.53
0.60	1.00	0.65	249.80
0.75	1.00	0.53	222.56
Smith-Waterman Threshold	**Precision**	**Recall**	**Total Time (s)**
0.10	0.93	0.93	1,754.78
0.20	1.00	0.80	1769.27
0.25	1.00	0.80	2281.85
0.40	1.00	0.53	2296.37
0.60	1.00	0.47	1804.24
0.75	1.00	0.40	1743.80

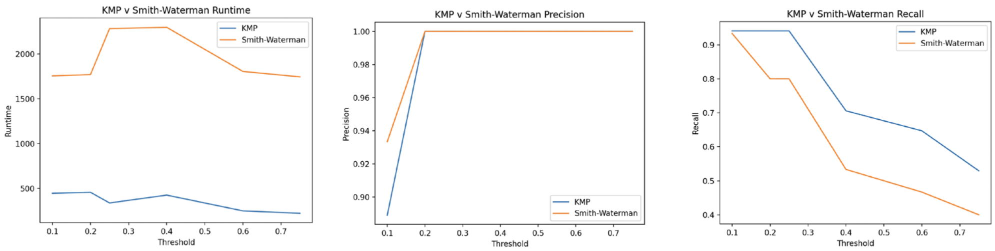


Figure 2. The KMP is the blue line and the Smith-Waterman is the orange line. These graphs compare the runtime, precision, and recall between the Smith-Waterman and KMP algorithms.

Discussion

We have explored the issue of melodic plagiarism detection. Our primary objective was to compare the performance of two algorithms, Knuth-Morris-Pratt (KMP) and Smith-Waterman, in identifying melodic plagiarism within music compositions. We achieved this by adapting algorithms originally designed for sequence alignment and string searching, repurposing them for the specific task of identifying analogous melodies in music. Our results [Table 1, Figure 2] demonstrate that KMP, an exact string searching algorithm, exhibits superior recall and efficiency compared to Smith-Waterman, a local sequence alignment method, across various similarity thresholds. This method offers a reliable and efficient tool for protecting the integrity of musical compositions. However, our study is not without its limitations. One significant limitation is the relatively small dataset of 50 data points used in the experiments. While we aimed to create plagiarized data by infusing parts of one melody into another, our dataset may not fully represent the variability and complexity of real-world music compositions. The runtime of smith-waterman is O(mn) which makes the runtime long due to the input being every single note. Additionally, the study primarily focuses on melodic similarities, ignoring other crucial aspects of music such as harmony and chord progression. Addressing these limitations presents opportunities for future work. To enhance the robustness of the algorithm, the dataset can be expanded to include a more diverse range of music compositions and explore the integration of additional features, like harmony analysis. Furthermore, research into the application of machine learning techniques to improve detection accuracy and developing a user-friendly software tool for music professionals would be promising avenues for future research in this field.

Conclusion

In summary, our research demonstrates the efficacy of repurposed algorithms, particularly Knuth-Morris-Pratt (KMP), in detecting melodic plagiarism within music compositions. The results highlight KMP's superior recall, comparable precision, and faster runtimes compared to the Smith-Waterman algorithm. While acknowledging limitations, such as a small dataset and a focus on melodic similarities, our study provides a foundation for future research. Ultimately, our findings contribute valuable insights, offering a pathway for developing more effective and objective tools to protect the creative integrity of musical compositions in the face of plagiarism concerns.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

Siwek, S. E. (2007). The True Cost of Sound Recording Piracy to the U.S. Economy: IPI." RIAA.

Sisario, Ben. "Ed Sheeran Wins Copyright Case over Marvin Gaye's 'Let's Get It On.'" The New York Times, 4 May 2023, www.nytimes.com/2023/05/04/arts/music/ed-sheeran-marvin-gaye-copyright-trial-verdict.html.

Park, K., Baek, S., Jeon, J., & Jeong, Y.-S. (2023, August 30). *Music Plagiarism Detection Based on Siamese CNN*. hcisj.com. http://hcisj.com/data/file/article/2022080003/12-38.pdf

Sie, Mu-Syuan, et al. "Detecting and Locating Plagiarism of Music Melodies by Path Exploration over a Binary Mask." Computer Science & Information Technology (CS & IT), Aug. 2017, https://doi.org/10.5121/csit.2017.71004. Accessed 30 Sept. 2023.

He, Tianyao, et al. Music Plagiarism Detection via Bipartite Graph Matching. arxiv.org/pdf/2107.09889.pdf.

Malandrino, Delfina, et al. "An Adaptive Meta-Heuristic for Music Plagiarism Detection Based on Text Similarity and Clustering." Data Mining and Knowledge Discovery, vol. 36, no. 4, May 2022, pp. 1301–34, https://doi.org/10.1007/s10618-022-00835-2.

Wikipedia contributors. "Knuth–Morris–Pratt Algorithm." *Wikipedia*, 31 Oct. 2023, en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm.
"Python - MIDI Collection." *Cymatics.fm*, cymatics.fm/products/python-midi-collection.

Unison Audio Inc. "Unison Beatmaker Blueprint (Free Teaser Pack) - Unison." *Unison Audio Inc.*, 1 June 2023, unison.audio/product/beatmaker-blueprint-free-teaser-pack.

"Unison Essential MIDI Melodies -." *Unison Audio Inc.*, 2 Jan. 2022, unison.audio/product/unison-essential-midi-melodies.rg/10.47611/jsr.v8i2.775