UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

Application of Change Point Detection Algorithms in Adaptable Symbolic Music Segmentation Task Using MIDI Representation

Sakib Mukter

Master thesis in Applied Computer Science … May 2024

“It always seems impossible until it is done.”
–Nelson Mandela

1

Abstract

This thesis studies the application of Change Point Detection (CPD) algorithms to the segmentation of symbolic music using MIDI representations. This study focuses on the use of two primary CPD algorithms, PELT and Binary Segmentation, to analyze and detect transitions within the Lakh MIDI dataset, which is known for its diversity in musical genres and styles. The main focus of the thesis is on how accurately the Binary Segmentation and PELT algorithms detect structural changes. The effectiveness of these algorithms is measured through precision, recall, and F1 score metrics, derived from manually annotated segments serving as ground truth. The adaptability of these methods across various musical structures is also evaluated to ensure their robustness and flexibility in handling different musical forms. The findings indicate that while both algorithms perform effectively, PELT shows superior adaptability and accuracy in segmenting musical structures.

2

Acknowledgements

I would like to express my profound appreciation to my supervisors, Professor Bernt Arild Bremdal and Shayan Dadman, for their invaluable guidance not only during this thesis project but throughout every Artificial Intelligence (AI) course during my master's studies. Additionally, heartfelt thanks to Professor Rune Dalmo for his openness and support throughout my master's journey.

Further, I want to extend my gratitude to my classmates, whose support was important in overcoming the challenges from the first term and beyond.

Finally, I owe immense gratitude to my parents and wife for their unwavering love, support, and guidance. Their encouragement to pursue and complete this degree has been a fundamental pillar of my success.

Table of Contents

List of Figures

List of Tables

3

Introduction

Music segmentation is an important element in the rapidly growing field of Music Information Retrieval (MIR), connecting musicological ideas and computer techniques. This thesis explores symbolic music segmentation, a field that seeks to split structured musical data, such as MIDI files, into meaningful segments or sections. These segments often sync up with unique musical concepts, which are essential for a wide range of uses, including recommendation systems, interactive music experiences, music analysis, and composition. This thesis focuses on the application and adaptation of Change Point Detection (CPD) algorithms in symbolic music segmentation. CPD algorithms, traditionally used in various fields such as economics, environmental science, and bio-informatics, are designed to identify points in time-series data where statistical properties exhibit significant shifts. Applying CPD algorithms to music segmentation involves detecting specific musical feature transition points, such as pitch patterns, rhythm patterns, or dynamic shifts, that define unique musical segments. However, the application of CPD to music data presents unique challenges. Unlike many time-series datasets found in other fields, music inherently illustrates a complex pattern of expressive nuances, cultural contexts, and theoretical constructs. Addressing this complexity could significantly enhance how music is segmented, leading to improvements in music recommendation engines, and enabling more nuanced music education tools. The choice of MIDI as the medium for this study is motivated by its prevalent use in music production, education, and research. MIDI provides a symbolic representation of music, encoding notes, timings, velocities, and instrumental assignments, among other parameters. This format offers a structured yet expressive framework for music, making it a perfect fit for computational analysis through CPD algorithms. For composers and musicians, improved segmentation could aid in the creative process by providing clearer analyses of music structure. Additionally, for academic and commercial music applications, better segmentation might lead to more engaging and personalized user experiences. In this thesis, we will explore the theoretical foundations of CPD algorithms, their current applications in the musical domain, and the methodologies for adapting these algorithms to the precision of symbolic music data in MIDI format. The objective is to advance the field of MIR by developing CPD-based techniques that not only efficiently segment symbolic music data but also resonate with the fundamental musical structures.

4 Background

In the field of Music Information Retrieval (MIR), the segmentation of symbolic music data, like MIDI files, into unique segments plays an important role. This literature review section discusses significant research studies that have addressed the segmentation and comprehensive analysis of symbolic music data. It aims to highlight the methodologies, findings, and technological advancements that have shaped my understanding and capabilities in this area. This review helps to see what has been done before and points out where more work is needed, setting up the possibility for new ideas and advancements in Music Information Retrieval (MIR).

4.1 Symbolic Music Structure and Segmentation

The fundamental concepts of music creation have been found by lots of research on the structure of symbolic music, which includes the grouping of musical ideas into a logical sequence. Analyzing symbolic music structure is similar to analyzing grammar in a language, where hierarchical connections are important. The development of computational models that aim at identifying and classifying structural features in symbolic music data benefited greatly from this perspective. [1] In symbolic music, segmentation is the technique of splitting a piece of music into segments that stand in for unique musical ideas or phrases. Researchers have developed various computational approaches to tackle the challenges of music segmentation. Techniques range from rule-based systems that use established music theory principles to identify segment boundaries, to data-driven approaches that employ machine learning algorithms to learn segmentation patterns from a large corpus of annotated music. The importance of limit detection and pattern recognition based on properties such as pitch, duration, and dynamics has been highlighted. [**2**] Machine learning models, especially those using supervised learning can be trained on annotated datasets to predict segment boundaries. In [**3**], he proposed the Local Boundary Detection Model (LBDM), which uses changes in melodic contour, rhythm, and other musical features to detect boundaries. This model has been influential in subsequent research on music segmentation. Hidden Markov Models (HMMs) have been used to model the probabilistic structure of music, treating segments as states and musical events as observations. Raphael and Stoddard used an unsupervised HMM for segmenting Bach's chorales, demonstrating the model's effectiveness in capturing repetitive structures. [**4**] An ensemble of temporal prediction error models that predict the next token during training to detect phrase changes, and employ a peak detection algorithm at test time to refine segment candidates. Their method demonstrated state-of-the-art performance on the Essen Folksong dataset, achieving impressive F-scores and R-values in an unsupervised setting. [**17**] However, my thesis specifically focuses on applying Change Point Detection (CPD) algorithms using MIDI representation to achieve adaptive segmentation of symbolic music, aiming to handle the variability in music across different genres and styles more effectively.

4.2 MIDI Representation in Music

MIDI (Musical Instrument Digital Interface) files offer a detailed representation of music, encoding not only the notes played but also the timing, duration, and intensity of each note. This form of symbolic representation is pivotal for MIR tasks as it provides a structured and adaptable format for analysis. The digital format of MIDI allows for intricate manipulations and analysis of musical components such as pitch, velocity (intensity), and timing (duration), which are critical for understanding the nuances of musical expression. Tzanetakis and Cook [**18**] have pointed out the utility of MIDI in providing a clear distinction between different instruments and notes for polyphonic music analysis. Jiang's thesis [**19**] provides an excellent example of the advantages of MIDI file usage in music analysis. In contrast to audio files, which need complex preprocessing to extract musical components, MIDI files offer direct access to precise data for every note played. This allows researchers to apply sophisticated algorithms directly to the music data, enhancing the accuracy of tasks such as melody tracking, chord detection, and structural analysis. Additionally, because MIDI data is organized, researchers may use it as a foundation for developing highly accurate algorithms that can learn and predict music segmentation. In my thesis, I utilize the Lakh MIDI dataset [**10**], which notably lacks annotations. I focus on exploiting musical properties such as pitch, velocity, duration, and timeshifts to develop a model for music segmentation.

4.3 CPD Algorithms in Music Segmentation

Change Point Detection (CPD) algorithms are important in various fields, including statistics, economics, environmental science, and particularly in music information retrieval (MIR). Change point detection algorithms are commonly categorized into two types: offline and online. Offline methods identify change points in the sequence by analyzing the full dataset in a single batch. Usually, the goal of this method is to identify every change point in a dataset once all the data has been gathered. In contrast, online, or real-time, algorithms execute alongside the ongoing process they are monitoring. They analyze each data point as it arrives with the objective of identifying changes as quickly as they occur, ideally before the arrival of the next data point. This method allows for immediate response and adjustment based on real-time data insights.[**5**] These methods include both supervised and unsupervised approaches, to develop the algorithm's expected result.

Supervised CPD algorithms rely on labeled datasets to learn the characteristics of change points. These algorithms typically employ models that have been trained to recognize the exact moments when changes occur based on previous examples. In [**6**] they proposed an automatic segmentation technique that merges Support Vector Machine (SVM) classification and self-similarity segmentation. They utilized a dataset of music where each file, ranging from 5 to 10 seconds in duration, was manually labeled. Advancements in neural networks, especially deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have opened new avenues for CPD in music segmentation. Böck and Schedl (2012) demonstrated the use of RNNs for real-time beat tracking and segmentation, leveraging the network's capability to process sequential data and learn complex patterns. [**21**]

Unsupervised CPD algorithms do not require labeled data, making them suitable for scenarios where annotations are unavailable or incomplete. These algorithms typically analyze the data to find inherent patterns or changes. In paper [**9**], they proposed an unsupervised method that is divided into three phases: adaptive normalization, recursive singular spectrum analysis, and change-point detection, respectively which refers to a deep neural network-based adaptive approach for detecting changes in multivariate time series data. A recent paper, [**7**] explored this through graph-based algorithms, introducing innovative methods like Norm, G-PELT, and G-Window for music segmentation. The Norm method relies on normalizing musical features such as Inter-Onset Intervals (IOIs) and pitch directions, constructing a vector that represents these features. This vector then forms a self-similarity matrix, helping to pinpoint segment boundaries by identifying peaks in the novelty curve. The G-PELT method takes this a step further by transforming music into a graph and utilizing the Pruned Exact Linear Time (PELT) algorithm to seek out significant structural changes, optimized through a cost function mindful of penalties to prevent overfitting. Alternatively, the G- Window method employs a sliding window technique over the graph's adjacency matrix to discern discrepancies in the music structure, marking potential segments. Tested on the Schubert Winterreise Dataset (SWD) and the Beethoven Sonatas Dataset (BPS), G-PELT emerged as particularly effective, with high recall rates indicative of its proficiency in detecting true structural boundaries. The change Point Detection (CPD) algorithms need to be able to detect not only quick statistical changes but also the more nuanced and unique changes that describe musical form and rhythm. Based on this idea, [**20**] proposes unique techniques for novelty detection that detect changes in melody, offering the first steps toward managing the statistical accuracy of CPD with the minor changes that come with music. In my thesis, I applied an unsupervised approach using an unlabeled dataset. Unlike their approach in [**7**], I build my novelty graph on features derived from Principal Component Analysis (PCA). As Nawal discussed in his blog [**8**], PCA efficiently extracts a low-dimensional set of features from a high-dimensional dataset, aiming to retain as much information as possible. This reduction not only simplifies the data but also enhances the clarity and significance of visualizations. Utilizing these PCA-derived features, I then applied Change Point Detection (CPD) algorithms to the resulting novelty curve, enabling effective segmentation of the music.

5 Methodology

The proposed method consists of three main components: i) A preprocessing step where MIDI files are parsed and relevant musical features are extracted; ii) Feature reduction using Principle Component Analysis (PCA) and iii) a segmentation step that applies a Change Point Detection (CPD) algorithm to the novelty curve derived from the PCA of the tokenized data. These components in Figure: 5.1, work together to preprocess, reduce dimensions, and segment the MIDI files, allowing for accurate analysis of musical structures.

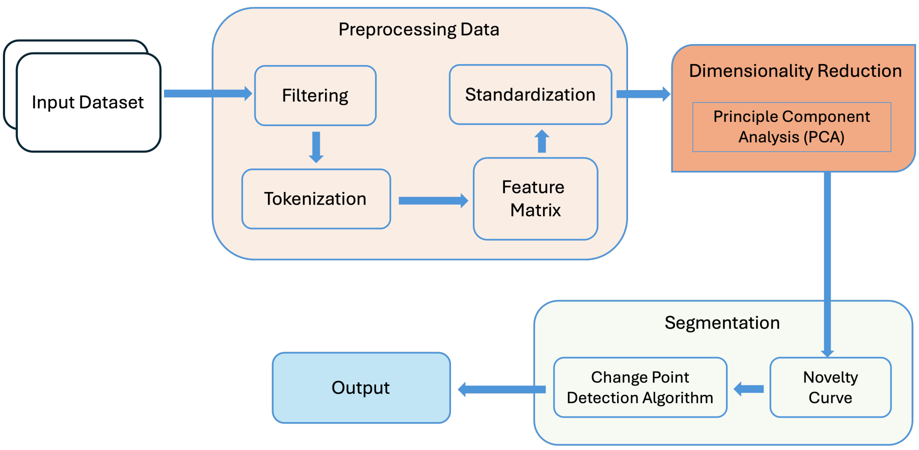

Figure 5.1: Proposed CPD-based Music Segmentation

5.1 Preprocessing

The selection of datasets, cleaning and filtering of the datasets, and feature extraction from the datasets are discussed in this section.

The dataset used for this study is the Lakh MIDI Dataset(LMD)[**10**], specifically the Lakh MIDI Dataset Matched(LMD-M). The Lakh MIDI Dataset is a comprehensive collection of MIDI files. Each MIDI file represents a musical composition, providing pitch, velocity, duration, and instrument data, which serve as inputs for this study. The Lakh MIDI dataset consists of approximately 176,000 MIDI files. The Lakh MIDI Dataset Matched (LMD-M) is a subset of LMD containing MIDI files matched to songs from the Million Song Dataset (MSD). [**11**]

The Lakh MIDI Dataset Matched (LMD-M) contains around 45,000 MIDI files, which can be challenging to process efficiently without the implementation of rigorous preprocessing steps to clean and filter the MIDI data before analysis. Before extracting the features from the MIDI files, They are filtered based on time signature and length. As mentioned in Algorithm 1, any file with a time signature different from 4/4 or a maximum tick count below 10 times the ticks per beat is excluded.

Algorithm 1 Check if a MIDI file is valid for analysis

```
function MIDI_VALID(midi)
    for each time signature ts in midi.time_signature_changes do
        if ts.numerator ≠ 4 then
            return false
        end if
    end for
    if midi.max_tick < 10 × midi.ticks_per_beat then
        return false
    end if
    return true
end function
```

For the feature extraction, the MidiTok library has been used. This library is specifically designed to tokenize MIDI files into structured sequences that can be used for various tasks involving music generation, classification, or analysis. [**12**] This library offers several tokenization models. For this research study, the structured model is particularly notable for its ability to handle MIDI sequences in a way that preserves detailed musical information across several dimensions. It breaks down MIDI events into four primary types of tokens: Pitch Tokens, Velocity Tokens, Duration Tokens, and Time-Shift Tokens. [**13**].

For tokenization, the filtered MIDI files are parsed to extract a list of note events using the MidiTok library. Tokenized data has been stored in JSON files for each MIDI file in the dataset. Figure: 5.2, shows the formation of the Structured Tokenization Model and how pitch, velocity, duration, and time-shift are stored in an order.

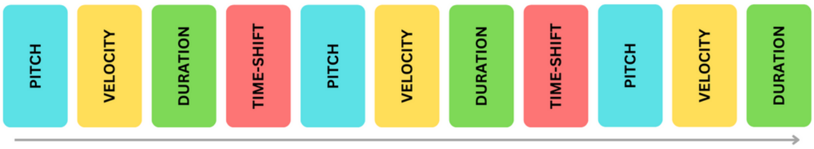

Figure 5.2: Structured Tokenization Model Formation

For retrieving the contents from the JSON files, the task is to extract specific musical attributes from these tokens, which are categorized into pitches, velocities, durations, and time shifts. As they are in order, so every fourth element starting from the 0th index is pitch tokens, the 1st index is velocity tokens, the 2nd index is duration tokens and the 3rd index is time-shift tokens.

The extracted features are compiled into a feature matrix, which organizes the data into a format suitable for machine learning and data analysis. This matrix combines the separate lists of pitches, velocities, durations, and time shifts into a single array where each column represents a specific feature and each row corresponds to a musical event (note) in the MIDI file.

Due to the differing ranges and scales of the musical features (e.g., pitch values are typically much higher numerically than durations or velocities), standardization is applied to normalize the data. This normalization ensures that each feature contributes equally to analytical models, preventing features with larger scales from disproportionately influencing the results. Standardization is achieved using the $StandardScaler$ from $Scikit-learn$ [14], which subtracts the mean and scales each feature to unit variance. The standardized feature matrix is now ready for further analysis. This normalization allows for the unbiased discovery of patterns and structures in the dataset, essential for the music segmentation tasks outlined in this study.

5.2 Dimensionality Reduction

Principal Component Analysis (PCA) is used as a dimensionality reduction technique. This step is important as it simplifies the data structure by reducing the number of variables considered while preserving the essential information that contributes most to the variance in the data. The primary objective of applying PCA is to transform the high-dimensional feature space (derived from MIDI files) into a lower-dimensional space that captures the most significant features necessary for understanding musical structure and dynamics. PCA is performed on the standardized feature matrix. The steps include the calculation of the covariance matrix to understand how feature dimensions vary from the mean with respect to each other. Then decompose the covariance matrix to its eigenvalues and eigenvectors. The eigenvectors represent the directions of maximum variance (principal components), and eigenvalues denote the magnitude of these variances. Then select a subset of principal components that capture a significant amount of variance in the data. This decision is often guided by the explained variance ratio, which quantifies the significance of each principal component. The original high-dimensional data is projected onto the space defined by the selected principal components. This results in a new dataset where each datum is now described by fewer dimensions that are most informative. PCA from $Scikit-learn$ [14] has been used to run the analysis in this study. It is highly optimized for performance, making it suitable for handling large datasets typical in MIR.

5.3 Implementation of Change Point Detection Algorithm

To implement a change point detection algorithm, a novelty curve needs to be derived from the PCA. The novelty curve serves as a pivotal component in the detection of structural changes in music. The steps to construct the novelty curve include the computation of the first-order difference of the PCA-transformed components. This difference highlights variations between consecutive analysis frames in the reduced feature space, emphasizing changes in the musical features. The next task is taking the absolute value of these differences to ensure that all changes contribute positively to the novelty measure, emphasizing the magnitude of change regardless of direction. Finally, sum these absolute differences across all principal components. This summation provides a single novelty value per time step, which aggregates the changes across all considered features. The novelty curve is ready for the change point detection algorithm.

Change point detection (CPD) is designed to pinpoint moments where there is a significant shift in the statistical characteristics of a data sequence. In this research, two CPD algorithms were deployed: PELT (Pruned Exact Linear Time) and Binary Segmentation. These methods are used to effectively detect and analyze variations within the data.

The PELT (Pruned Exact Linear Time) algorithm is a change point detection method that handles the computational challenges typically associated with segmenting large data sets. The underlying principle of the PELT algorithm is the identification of optimal change points in a time series, which involves evaluating various possible segmentations of the data. Considering the challenge of listing every possible split related to the data size's exponential growth, PELT implements a dynamic programming approach combined with a pruning rule. This rule is important because it reduces the computational load of the method by allowing it to ignore indexes that have a very small probability of having a change point. The computational efficiency of PELT also depends on the frequency of evaluating the cost function, which measures the fit of the model to segments of the data. The pruning effectively reduces the number of times this costly computation is required. the average computational complexity is of the order of $O(CKn)$, where K is the number of change points to detect, n is the number of samples, and C is the complexity of calling the considered cost function on one sub-signal.[**15**] The cost function Least square deviation(CostL2) has been used in this algorithm. detects mean shifts in a signal. The mean-shift model is among the earliest and most extensively researched models in the change point detection field, as noted in various studies. This model uses a Gaussian distribution with a consistent variance. Formally, for a signal y_{t_t} on an interval I,

$$c(y_I) = \sum_{t \in I} ||y_t - \bar{y}||_2^2 \tag{5.1}$$

where $\bar{y}$ is the mean of $\{y_t\}_{t \in I}$ [**16**]

Binary Segmentation, commonly referred to as Binseg, is a robust and efficient algorithm for detecting multiple change points in complex datasets. This method is particularly well-suited to applications in music information retrieval where the data might exhibit multiple structural breaks. The mechanism of Binseg starts by identifying a single change point across the entire dataset first. Upon detecting this change point, the dataset is split into two sub-segments at the identified point. This process is recursively applied to each resulting segment until no further significant change points are detected or a predefined criterion is met. The advantages of binary segmentation include its low computational complexity, which is $O(Cnlogn)$, where n represents the number of samples and C is the complexity of executing the cost function on an individual sub-signal. This linear complexity makes Binseg particularly advantageous for handling large datasets, as it ensures that the segmentation process remains computationally feasible even as the size of the data increases.

Additionally, this method can enhance any single change point detection tech-

Algorithm 2 Symbolic Music Segmentation Using CPD

```
Input: MIDI file path
Output: Change points in the music data
procedure MusicSegmentation(file_path)
    data ← READ_FILE(file_path)
    normalized_data ← DATA_NORMALIZATION(data)
    pca_result ← APPLE_PCA(normalized_data)
    novelty_curve ← NOVELTY_CURVE_GENERATE(pca_result)
    change_points ← ApplyCPD(novelty_curve)
    return change_points
end procedure
```

nique to identify multiple change points. It is also flexible in that it does not require prior knowledge of the number of regimes to function effectively. The *ruptures* library has been used to implement the change point detection algorithms in this study. [**16**] In Algorithm 2, the *applyCPD* method calculates both PELT and BinSeg algorithm and returns the change points within the MIDI data.

5.4 Tools

In this thesis, several key tools and technologies were used to effectively handle the segmentation of symbolic music using MIDI files through Change Point Detection (CPD) algorithms. The breakdown of the primary tools used is as follows.

5.4.1 Programming Language

Python is used as the primary programming language for implementing and organizing the workflow of data preprocessing, analysis, and segmentation. Python's extensive libraries and its supportive community make it an excellent choice for data analysis, machine learning, and scientific computing.

5.4.2 Platform

Selecting Jupyter Notebook was a good idea as it provided an interactive environment for coding, visualizing, and immediate result checking, which was invaluable for iterative testing and documentation of the analysis process.

5.4.3 Libraries

The *MidiTok* library was handy for tokenizing MIDI files into structured sequences. *MidiTok* facilitated the extraction and handling of detailed musical features such as pitch, velocity, duration, and time shifts, which are important for the analysis. The $scikit - learn$ library was used for data normalization and Principal Component Analysis (PCA), simplifying the process of reducing data complexity and improving interpretability. Additionally, the *Ruptures* library was used for change point detection. This library supports a variety of algorithms, including PELT and Binary Segmentation, which were integral to this thesis. Ruptures provided efficient methods for analyzing and identifying significant changes within the structured musical data.

6

Results

In this section, the results of the Principal Component Analysis (PCA) and subsequent Change Point Detection (CPD) algorithms applied to the Lakh MIDI dataset are presented. The PCA successfully reduced the dimensionality of the dataset, encapsulating the majority of the variance within the first three principal components. The loadings of the PCA revealed significant contributions from pitches, velocities, durations, and time shifts, each varying in influence across the components. Following PCA, the application of CPD algorithms identified distinct structural changes within the music compositions.

Table 6.1: PCA Loadings for Musical Features

Feature	PC1	PC2	PC3
Pitches	0.536988	0.237223	0.758393
Velocity	-0.539783	0.408238	0.467023
Duration	-0.173642	-0.876402	0.436566
Time Shifts	0.624601	-0.094791	-0.127042

In table 6.1 PC1 appears to be influenced positively by Pitches and Time Shifts but negatively by Velocity and Duration. This suggests that PC1 may represent a balance of these features where the overall pitch and timing between events are countered by how quickly and for how long the notes are played. PC2 is most strongly defined by Duration in a negative direction, with lesser but still significant contributions from Velocity. This could imply that PC2 encapsulates aspects of the music where sustained notes (long durations) and dynamic variability (velocity) play critical roles. PC3 is most influenced by Pitches and less so but still significantly by Velocity and Duration. The impact of pitch on PC3 could indicate a component that captures melodic dynamics.

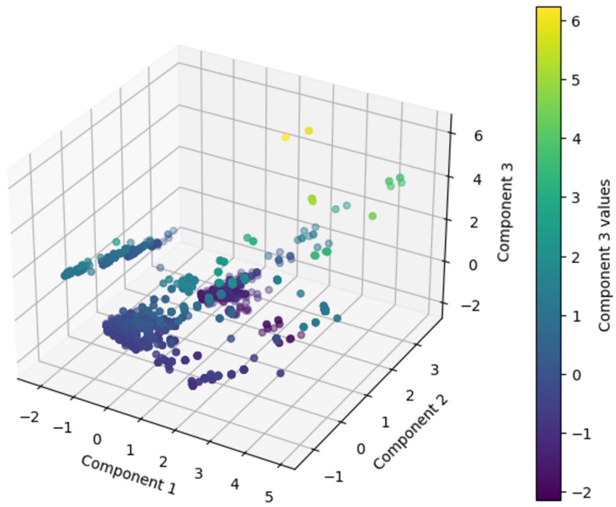

Figure 6.1: Visualization of PCA Components

In Figure 6.1, the axes of the plot are labeled as Component 1, Component 2, and Component 3, which represent the first, second, and third principal components derived from PCA. These components are linear combinations of the original variables that capture the greatest variance in the data set. Principal Component 1 (X-axis) likely captures the highest variance, followed by Principal Component 2 (Y-axis), and Principal Component 3 (Z-axis) captures the third highest. Each point on the plot corresponds to an observation (or data point) in the original dataset, projected into the space defined by the first three principal components. The coloring of the data points ranges from violet to yellow, which corresponds to the values of Component 3 (as indicated by the color bar on the right). This suggests that the color scale is used to visually enhance the differentiation based on the third principal component's values.

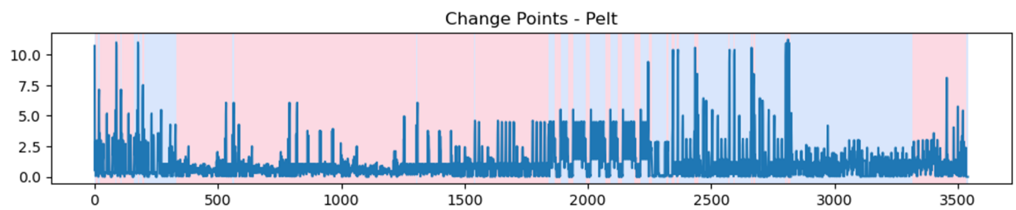

Figure 6.2: Change Point Detection using PELT

In Figure 6.2 and Figure 6.3, the red vertical lines are the markers for detected change points where significant changes in data properties are recognized by each algorithm. The pink shaded areas are the intervals between detected change points, implying segments where the data properties are relatively stable according to the algorithm.

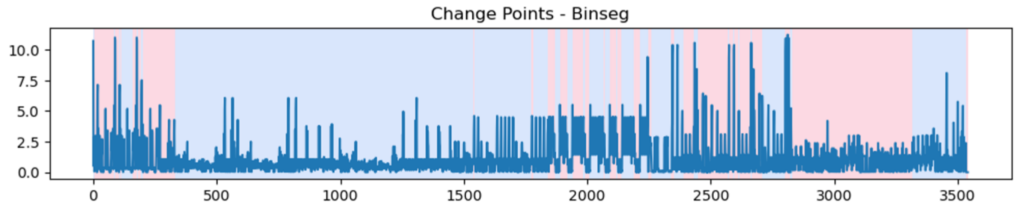

Figure 6.3: Change Point Detection using Binary Segmentation

7 Discussion

The findings of this thesis demonstrate the potential of unsupervised Change Point Detection (CPD) algorithms in effectively segmenting musical structures within MIDI files. The main objective of this thesis was to segment symbolic music using the Change Point Detection (CPD) algorithm with MIDI representation. I choose the unsupervised method because it may handle a variety of different situations without requiring prior training for each situation.

The ground truth boundaries for the selected Lakh MIDI dataset were not available. To address this, I manually created ground truth annotations by listening to several MIDI files and marking significant transitions and structural changes. These manually annotated files were then used to evaluate the performance of the Change Point Detection (CPD) algorithm. From Table: 7.1, the PELT algorithm shows a moderate balance between recall and precision, suggesting that while it can detect true change points with a certain level of accuracy, it also misses several actual changes and produces false positive results. The slightly higher recall compared to precision suggests it is more comprehensive in attempting to detect changes, though many of the detected points are not true changes (lower precision). The Binary Segmentation (Binseg) algorithm shows higher precision than PELT, indicating that a greater proportion of the change points it detects are true positives. However, the recall is significantly lower, which suggests that while the change points it identifies are likely to be correct, it misses a large number of actual changes. This results in many true change points not being detected by the algorithm. One of the reasons for low recall in Binseg could be the manual annotations of the data. As it initially divides the data into two segments at the most significant change point and then continues recursively, this approach might overlook smaller or more subtle changes that are nonetheless significant in the context of music segmentation. The F1 score is calculated using the precision and recall values to provide a measure that balances both metrics.

$$F1 = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

The F1 Score for PELT is approximately 40.95 and Binseg is approximately 23.70. These F1 scores provide a more balanced view of each algorithm's performance, factoring in both precision and recall. The higher F1 score for PELT indicates it is more effective overall in this particular evaluation setting compared to Binseg, which showed significantly lower effectiveness, particularly in recall, leading to its lower F1 score.

Table 7.1: Proposed CPD Algorithms Performance

Algorithm	Precision	Recall	F1 Score
PELT	39.63	42.33	40.95
BinSeg	38.01	17.22	23.70

In Figure 6.2 and Figure 6.3, the change points marked by the PELT algorithm are somewhat evenly distributed throughout the dataset but with noticeable clusters around certain regions. This might suggest that PELT is sensitive to changes that occur with medium to high frequency throughout the data series. In contrast, the Binseg algorithm appears to have more uniformly distributed change points throughout the entire range of the dataset. The distribution is a bit more spread out than in the PELT graph.

Table 7.2: Number of Detected Change Points by PELT and BinSeg at Various Noise Levels

Noise Level	PELT	BinSeg
0	57	54
0.05	59	54
0.1	58	53
0.2	63	48
0.5	73	51

In Table 7.2, the noise level increases, and the number of detected change points by PELT generally increases from 57 at no noise to 73 at a noise level of 0.5. Also

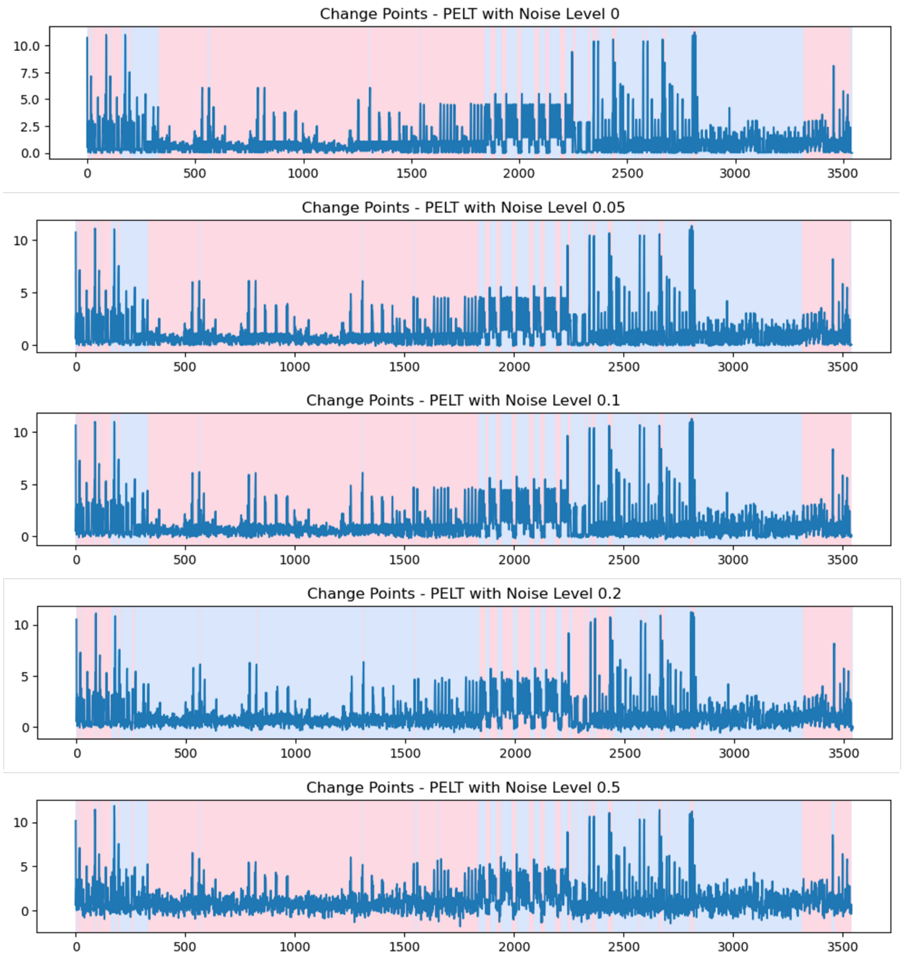

Figure 7.1: PELT in Different Noise Levels

Figure 7.1 and 7.2 shows that PELT might be more sensitive to noise, detecting more potential change points as the data becomes more varied and possibly more complex. In contrast, the BinSeg algorithm shows a different pattern, with the number of detected change points decreasing as noise increases, particularly noticeable from a noise level of 0.2 and beyond. This might indicate that BinSeg is less robust to high noise levels, perhaps missing change points due to increased data variability.

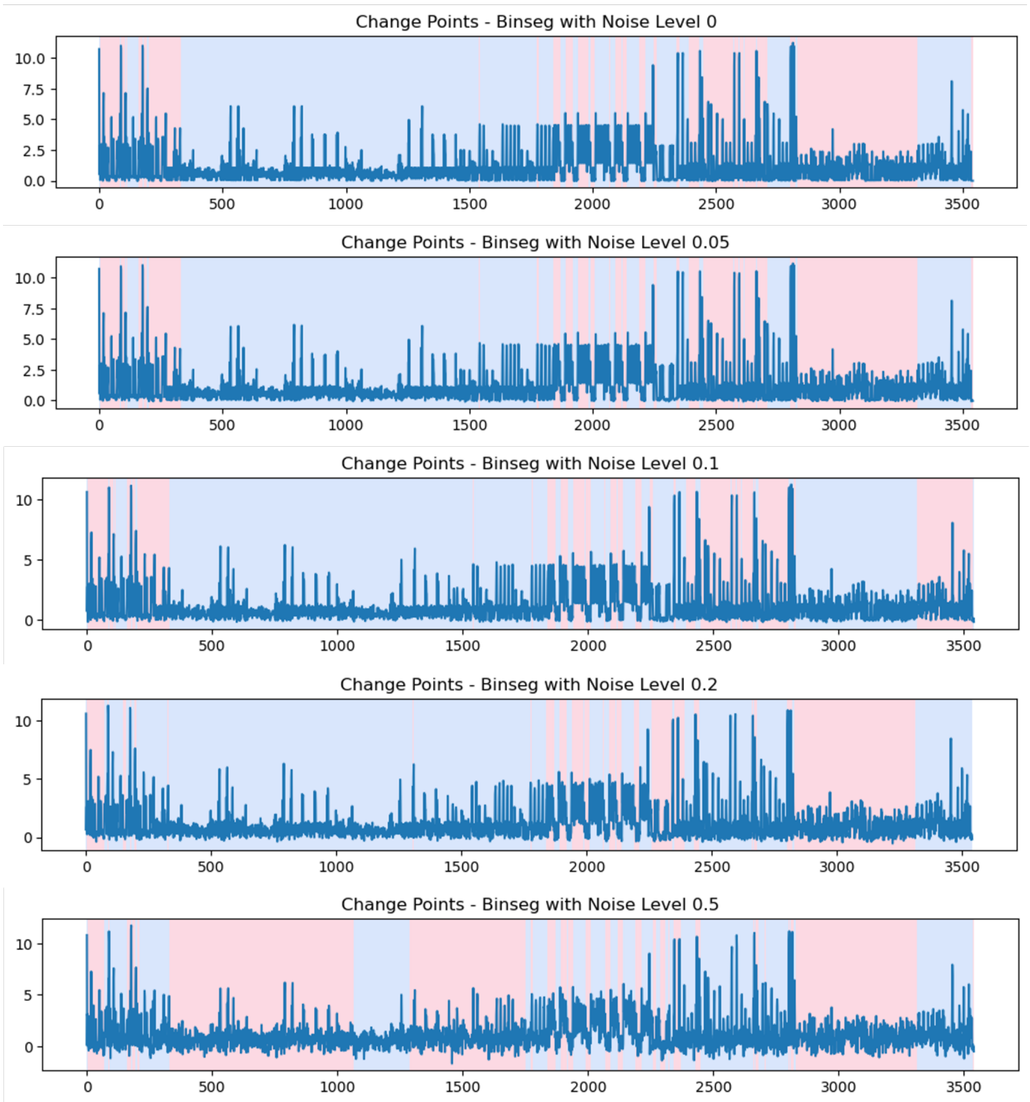

Figure 7.2: BinSeg in Different Noise Levels

8

Conclusion and Future Works

This thesis presented two techniques aimed at segmenting symbolic music within MIDI representation. Upon evaluating the performance of each technique using the Lakh MIDI dataset for symbolic structure segmentation, it becomes clear that the PELT change point detection algorithm works better than the BinSeg algorithm. This study shows that by adjusting the parameters of these algorithms, music can be segmented into various structural levels. This ability improves applications like music production and classification while increasing our knowledge of musical composition. The methods presented are both offline and unsupervised, making them suitable for processing extensive datasets. However, the results of this study may be improved by adding a learning algorithm for a more in-depth examination of particular musical genres. Future work will be included in the part that follows. It contains further studies and developments related to this research.

8.1 Future Work

The findings presented in the previous chapter indicate that the performance of the change point detection (CPD) algorithms did not meet expectations. A significant factor contributing to this outcome could be the reliance on manually identified ground truth data. Future research should consider accurate labeling because this is very important for the effective evaluation of CPD algorithms and incorporating more advanced machine learning models, such as deep learning architectures, which may enhance pattern recognition capabilities. Additionally, exploring semi-supervised learning techniques could prove beneficial. These methods use a limited set of labeled data to inform the unsupervised learning process, potentially improving the accuracy of CPD algorithms to a level closer to that of supervised methods.

Bibliography

[1] Temperley, D. (2001). The Cognition of Basic Musical Structures. MIT Press.

[2] Cambouropoulos, E. (2006). Musical Parallelism and Melodic Segmentation: A Computational Approach. Music Perception, 23(3), 249-269.

[3] The local boundary detection model (LBDM) and its application in the study of expressive timing E Cambouropoulos - ICMC,

[4] Christopher Raphael, Joshua Stoddard; Functional Harmonic Analysis Using Probabilistic Models. Computer Music Journal 2004

[5] Aminikhanghahi, Samaneh & Cook, Diane. (2017). A Survey of Methods for Time Series Change Point Detection. Knowledge and Information Systems. 51. 10.1007/s10115-016-0987-z.

[6] Li, Feng & You, You & Lu, Yuqin & Pan, YuQing. (2014). An Automatic Segmentation Method of Popular Music Based on SVM and Self-similarity. 15-25. $10.1007/978-3-319-15554-8_2$.

[7] Hernandez-Olivan, Carlos, Sonia Rubio Llamas, and Jose R. Beltran. Symbolic Music Structure Analysis with Graph Representations and Changepoint Detection Methods. arXiv preprint arXiv:2303.13881 (2023).

[8] All about the Music — 01 by Pratham Nawal `https://towardsdatascience.com/all-about-the-music-01-ad1b989260df`

[9] Muktesh Gupta, Rajesh Wadhvani, Akhtar Rasool. Real-time Change-Point Detection: A deep neural network-based adaptive approach for detecting changes in multivariate time series data, Expert Systems with Applications, Volume 209, 2022, 118260, ISSN 0957-4174

[10] Colin Raffel. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching. PhD Thesis, 2016.

[11] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

[12] Fradet, Nathan & Briot, Jean-Pierre & Chhel, Fabien & Seghrouchni, Amal & Gutowski, Nicolas. (2021). MIDITOK: A PYTHON PACKAGE FOR MIDI FILE TOKENIZATION.

[13] Hsiao, Wen-Yi & Liu, Jen-Yu & Yeh, Yin-Cheng & Yang, yi-hsuan. (2021). Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs. Proceedings of the AAAI Conference on Artificial Intelligence. 35. 178-186. 10.1609/aaai.v35i1.16091.

[14] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011

[15] Killick, R., Fearnhead, P., & Eckley, I. (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500), 1590–1598.

[16] Charles Truong, Laurent Oudre, Nicolas Vayatis, Selective review of offline change point detection methods, Signal Processing, Volume 167, 2020, 107299, ISSN 0165-1684, https://doi.org/10.1016/j.sigpro.2019.107299.

[17] Bassan, Shahaf & Adi, Yossi & Rosenschein, Jeffrey. (2022). Unsupervised Symbolic Music Segmentation using Ensemble Temporal Prediction Errors. 2423-2427. 10.21437/Interspeech.2022-10379.

[18] Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. IEEE Transactions on speech and audio processing.

[19] Jiang, Z. (2019). Automatic Analysis of Music in Standard MIDI Files [Carnegie Mellon University]. `https://www.cs.cmu.edu/~music/papers/zheng_jiang_thesis.pdf`

[20] Foote, Jonathan. (2000). Automatic audio segmentation using a measure of audio novelty. IEEE International Conference on Multi-Media and Expo. 1. 452 - 455 vol.1. 10.1109/ICME.2000.869637.

[21] Böck, S., & Schedl, M. (2012). Enhanced beat tracking with contextual neural networks. Proceedings of the International Conference on Digital Audio Effects (DAFx).