Bachelor's Expose: Implementing the Spikelet Algorithm into ClaSP: An Analysis of Performance Improvement

Victor Munacoha

Humboldt Universität zu Berlin, Rudower Ch 25, 12489 Berlin, Germany

1 Introduction

With today's ever-evolving technology, we have entered what many call the Age of Information. Collecting, analyzing, and evaluating data has become more important than ever, and its significance is unlikely to decrease in the foreseeable future. One of the key methods for representing data is Time Series analysis.

Time Series are a series of data points sorted in timed order. They have many practical applications in different fields, such as monitoring heart rates in healthcare or tracking stock prices and trading volumes in the financial market. One important form of Time Series Analysis is Time Series Segmentation (TSS)[1]. It is a way to find the underlying properties of a Time Series by dividing the dataset into a sequence of segments. A common use for TSS in audio signals is when they are partitioned into multiple segments based on who is speaking at that time[2].

However, when dealing with especially long or complex time series, correctly segmenting the data becomes increasingly difficult. The challenge lies in accurately identifying the boundaries where significant changes occur, while also managing the computational cost of processing large datasets. As time series become longer and more detailed, the time and memory resources required for analysis increase.

ClaSP (Classification Score Profile)[3] is an algorithm designed to address these challenges by transforming the time series segmentation problem into a binary classification task. ClaSP iteratively identifies change points by assessing the dissimilarity between subsequences of the time series, offering a fast and scalable solution[3]. While ClaSP has proven highly accurate in various domains, its performance can most likely be improved regarding runtime scalability and handling large datasets due to its quadratic time and memory complexity in the time series length.

This thesis proposes integrating the Spikelet Transformation[4, 5] into ClaSP to enhance its ability to process large time series efficiently. The Spikelet Transformation reduces data complexity by retaining critical points and discarding redundant information, thus simplifying the time series without losing the most essential features. By combining Spikelet's data reduction capabilities with ClaSP's segmentation accuracy, the goal of this thesis is to assess whether the approach improves runtime and memory scalability without compromising the quality of segmentation.

2 Research Goals

The primary objective of this thesis is to integrate the Spikelet Transformation into the ClaSP framework and analyze how this integration affects the performance of time series segmentation when the time series are especially long. This approach aims to reduce the computational and memory complexity by simplifying long time series while maintaining high segmentation accuracy.

2.1 Approach

The Spikelet Transformation decomposes a series into key segments, or "spikes", each characterized by magnitude and support length [4,5]. This adaptive approximation reduces data complexity by filtering out noise and retaining essential structural patterns, represented as convex, concave, or constant spikes (Fig.1). Using input parameters like Magnitude Threshold (MaT)—which controls the minimum spike size by filtering out smaller fluctuations to focus on significant features—and Constant Length Threshold (CoT)—which sets the minimum duration, a segment must have, to be considered constant, helping to balance detail and simplicity—the algorithm fine-tunes the level of detail, balancing noise reduction and pattern preservation. By simplifying time series data into patterns of key shapes, Spikelet makes it easier to find repeating structures, matching spikes' peaks and valleys for efficient large-scale analysis. Using the Spikelet Transformation as a pre-processing step prior to using ClaSP aims to enhance runtime and memory scalability, ideally allowing ClaSP to segment longer time series efficiently without compromising accuracy.

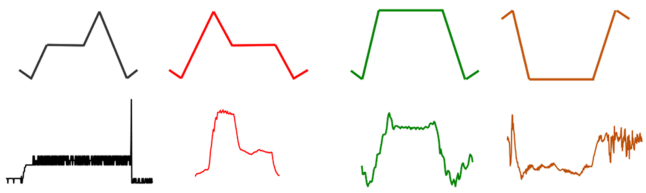

Fig. 1. Spikelet applied on segments of a time series

As a first step, the Spikelet code will be translated from MATLAB into Python, the language ClaSP is written in. Only the necessary parts for the Spikelet Transformation will be considered for the translation. The Transformation will be implemented right after loading the time series and as a preprocessing step prior to the ClaSP-Segmentation. Before starting with segmentation experiments, one dataset will be used to test the correctness of the Spikelet implementation by comparing the results of both MATLAB and Python implementations.

After fully implementing the Spikelet Transformation into ClaSP, segmentation experiments will be conducted with the intent to maximize the desired outcome by adjusting the Spikelet Magnitude Threshold, Spikelet Constant Length Threshold, and ClaSP Window Size—which determines the length of the subsequences and thus the change point granularity used in the segmentation process by ClaSP.

If the results with a constant MaT and CoT are not satisfactory, a dynamic adjustment of the parameters will be implemented and tested. The goal is to adjust the parameters in each segment fitting to the data itself so that if a time

series varies significantly between segments, the MaT and CoT will be adjusted accordingly. Using that method, the transformation will no longer be done before the ClaSP-Segmentation step, but during that step.

3 Evaluation

The evaluation of this research will focus on comparing the performance of the original ClaSP, the Spikelet-enhanced ClaSP, and, if necessary, the Spikelet-enhanced ClaSP with dynamic adjustment of Hyperparameters. This evaluation will assess how effectively the transformation improves runtime and memory scalability without compromising on the scoring.

To properly assess the evaluation, the algorithm will be tested using a set of established benchmark datasets[6], similar to those used in evaluating ClaSP. The focus will be on longer time series, since ClaSP is already efficient for shorter benchmarks[3], meaning, it primarily makes sense for long time series because those have a longer runtime which could benefit from the transformation. Each Benchmark will also be tested multiple times, to assure consistency and avoid outliers. By comparing the performance of the algorithm against the original segmentation tool, the study will provide a clear understanding of the strengths and weaknesses of the Spikelet-ClaSP integration.

3.1 Metrics

The evaluation will focus on the following key performance metrics:

Runtime and Memory Efficiency: The runtime and memory efficiency of the Spikelet-ClaSP algorithm will be compared against the original ClaSP to assess whether the integration of the Spikelet algorithm results in faster and more memory-efficient segmentation, particularly for long and complex time series datasets.

Scoring: This will measure how well the algorithm identifies change points in time series data compared to the correct points. A lower segmentation error (i.e., the difference between the predicted and actual change points) will indicate higher accuracy.

3.2 Result Analysis

The results will be analyzed to determine how well the Spikelet-ClaSP performs relative to the original ClaSP. The key focus will be on identifying any improvements in runtime and scalability, as well as any potential trade-offs in segmentation accuracy. By setting up the experiments in this manner, the study will provide a clear and objective assessment of the performance of the Spikelet-enhanced ClaSP algorithm.

References

1. Statistics Easily.: Was ist Zeitreihensegmentierung? Statistics Easily. Accessed on November 8, 2024. Available at: https://de.statisticseasily.com/Glossar/Was-ist-Zeitreihensegmentierung%3F/
2. T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A Review of Speaker Diarization: Recent Advances with Deep Learning," 2021, arXiv. doi: 10.48550/ARXIV.2101.09624.
3. Schäfer, P., Ermshaus, A., and Leser, U.: ClaSP - Time Series Segmentation. ACM CIKM 30(11), 1578–1587 (2021)
4. Imamura, M., Nakamura, T.: Parameter-free Spikelet: Discovering Different Length and Warped Time Series Motifs using an Adaptive Time Series Representation. Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2023)
5. Imamura, M., Nakamura, T.: Efficient Discovery of Time Series Motifs under both Length Differences and Warping. Proceedings of the 30th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2024)
6. Ermshaus, Arik, Patrick Schäfer, and Ulf Leser. "Raising the ClaSS of Streaming Time Series Segmentation." Proceedings of the VLDB Endowment 17.8 (2024): 1953-1966.