Latent Motif Discovery using Maximum Clique algorithms

Student Research Project Exposé

Leonard Clauß
Humboldt University of Berlin

November 28, 2020

Supervisors: Dr. rer. nat. Patrick Schäfer
Prof. Dr.-Ing. Ulf Leser

Contents

1 Introduction

A time series is a sequence of real valued numbers, e.g. recorded from a sensor, ordered in time. As the amount of available data increased drastically over the last few years, time series analysis gained a lot of attention in recent research. A typical challenge is motif discovery, i.e. the problem of finding frequently occurring patterns within a time series (Figure 1). It is an unsupervised learning problem. Motif discovery is used as an exploratory task across a multitude of domains, e.g. medicine [1], biology [2], meteorology [9] and robotics [13].

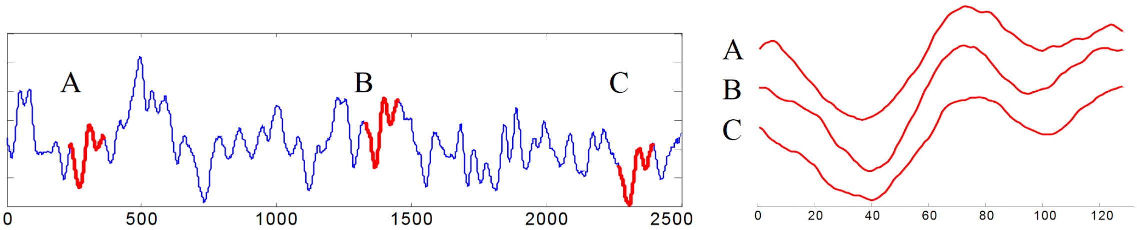

Figure 1: A time series contains a pattern that repeats three times (left). The direct comparison (right) shows that, except for the offset, the marked subsequences are very similar to each other.[1]

In literature, motifs are found in three different variations, which we will define precisely in the Background section. All these definitions assume that a similarity measure, e.g. Euclidean distance, and a subsequence (motif) size are given.

- *Pair Motifs*, as defined by Mueen et al. [11], are the pairs of subsequences in a time series that are most similar.
- *Set Motifs* are those subsequences in a time series that occur frequently, whereas only non-overlapping subsequences within a given similarity range are counted. These form a hypersphere. [7]
- *Latent Motifs* again are sequences that occur frequently in a time series, given a predefined similarity range. Two different definitions exist:
  - *Latent Learning Motifs* are similar in definition to set motifs and form a hypersphere. However, the sequence itself, i.e. the center of the motif, must not necessarily be a subsequence of the analyzed time series. [4]
  - *Latent Range Motifs* are sets of subsequences in a time series that are pairwise within the similarity range and thus form a Reuleaux polygon. [11]

Although many motif discovery methods were proposed, most of them address the pair motif problem. Only few search for set or latent motifs (see section 4.1 for an overview). Farther there is no exact discovery algorithm for latent motifs, as shown by Moczalla [10]. On top of that, we suspect that the problem of finding latent motifs is NP-hard. However, this will be shown in a future work.

Thus, we propose to examine a new approach for latent motif discovery. The main idea is to convert the pairwise subsequence distances (distance matrix) of a time series into a so-called distance graph. Given a motif size l, this graph will contain a node for

[1] figure from [14], page 1

each subsequence of length l. Two nodes are connected by an unweighted edge if their respective subsequences do not overlap and are within the given similarity range. The problem of finding latent motifs thus resolves to finding maximum cliques which exactly matches the definition of latent range motifs.

In the next section we will first of all show definitions of the previously mentioned terms. Afterwards, in Section 3 we will show that the latent range motifs are a superset to the latent learning motifs. Section 4 then gives an overview of the related work. The last two sections present the objectives and planned methods for this work.

2 Background

In the following we formally define the aforementioned variations of the motif discovery problem in literature.

Definition 2.1 (Time Series)**.** A time series $T = (t_1, t_2, ..., t_n)$ of length n is an ordered sequence of n real-valued numbers.

Definition 2.2 (Subsequence)**.** A subsequence $S_{i,l}$ in a time series $T = (t_1, t_2, ..., t_n)$ with $1 \leq i \leq n$ and $1 \leq l \leq n - i + 1$ is itself a time series of length l and defined as $S_{i,l} = (t_i, t_{i+1}, ..., t_{i+l-1})$.

Definition 2.3 (Overlapping subsequences)**.** Two subsequences $S_{i,l}$ and $S_{j,l}$ in a time series T overlap if and only if $i \leq j < i + l$ or $j \leq i < j + l$, i.e. they share at least one index of T.

For the following definitions assume a length l and a distance measure $d(\cdot, \cdot)$ on time series of length l are given. We are now able to define the first type of motifs.

Definition 2.4 (Top Pair Motif [11])**.** The top pair motif $P = \{S_{i,l}, S_{j,l}\}$ of a time series T is the set of two non-overlapping subsequences in T that have the minimal distance $d(S_{i,l}, S_{j,l})$ among all pairs of non-overlapping subsequences.

Note that there might be multiple different top pair motifs if they share the same (minimum) distance. Next we define set and latent motifs that describe repeating patterns in a time series.

Definition 2.5 (r-matching time series)**.** Two time series (or subsequences) T_1 and T_2 of length l are r-matching if and only if $d(T_1, T_2) \leq r$.

Definition 2.6 (Top Set Motif [7])**.** Given a radius r. The top set motif T_M of a time series T is the subsequence of length l in T that has the highest number of occurrences. That means, it has the largest set M of pairwise non-overlapping subsequences of length l in T that are r-matching to T_M.

Definition 2.7 (Top Latent Range Motif [11])**.** Given a radius r. The top latent range motif R of a time series T is the largest set of subsequences in T with length l that are pairwise non-overlapping and pairwise $2r$-matching.

Definition 2.8 (Top Latent Learning Motif [4])**.** Given a radius r. The top latent learning motif T_L of a time series T is a time series of length l, not necessarily a subsequence of T, that has the highest number of occurrences in T. That means, it has the largest set L of pairwise non-overlapping subsequences of length l in T that are r-matching to T_L.

Geometrically, the top set motif and top latent learning motif each form a hypersphere with radius r around T_M and T_L, respectively. The top latent range motif forms a Reuleaux polygon.

Again, there might be multiple different top set, latent range and latent learning motifs for a single time series. Finally, the pair, set, latent range and latent learning motif discovery problems are defined as finding their respective top motif.

3 Relation of Latent Motifs

Our first observation is that the solution to the latent range motif problem is a superset of the latent learning motif problem:

Lemma 3.1. *Given a time series T and a distance metric $d(\cdot,\cdot)$. For every latent learning motif T_L, its corresponding set of subsequences L is also a latent range motif of size $|L|$.*

Proof. $\forall S_1, S_2 \in L : d(S_1, T_L) \leq r \wedge d(T_L, S_2) \leq r \Rightarrow d(S_1, S_2) \leq 2r$ (triangle inequality). Thus L is a latent range motif of size $|L|$. □

Consequently, the size of the top latent learning motif subsequence set is a lower bound for the size of the top latent range motif.

Corollary 3.1.1. *Given the top latent range motif R and the top latent learning motif T_L and its corresponding set of subsequences L of a time series T. Then it always holds that $|R| \geq |L|$.*

On the other hand, a latent range motif must not always correspond to a latent learning motif. Figure 2 shows an example using the Euclidean distance as the distance metric d. The three two-dimensional subsequences S_1, S_2, S_3 form a latent range motif because $d(S_1, S_2) = d(S_1, S_3) = 2r$ and $d(S_2, S_3) = r$. However, there is no center point which is within distance r from each subsequence.

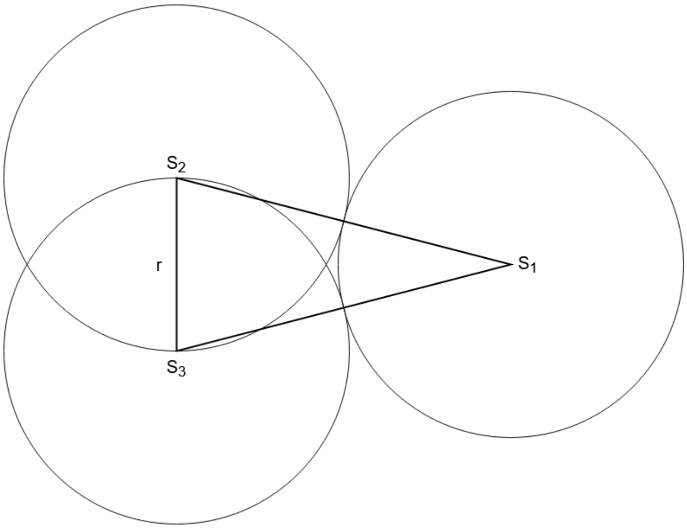

Figure 2: A latent range motif $\{S_1, S_2, S_3\}$ that does not correspond to a latent learning motif.

Because of this result, by providing an algorithm that finds the top latent range motifs and then filtering the results one is also able to find the top latent learning motifs.

4 Related Work

4.1 Set and Latent Motif Discovery

There are many pair motif discovery algorithms such as MK [11], but only few discover set or latent motifs. As for the former, the first method published was EMMA [14]. The algorithm first discretizes a time series and its subsequences into words using SAX [8]. They introduce a distance measure on this low-dimensional representation that is a lower bound for the Euclidean distance between two subsequences. The algorithm can therefore search for motifs in this discrete representation instead of in the real-valued time series and then filter false positives based on the real distances.

Another more recent method that is used for set and latent motif discovery is GrammarViz [17]. This algorithm again calculates the SAX representation of the subsequences. It then uses a compression algorithm to find rules of a context-free grammar for these words. The most used rules represent frequently occurring subsequences and therefore motifs.

Bagnall et al. [3] introduced three methods for set and latent range motif discovery, namely ScanMK, ClusterMK and SetFinder. The former two use the MK algorithm [11] to initially find the top pair motif. ScanMK then finds a latent range motif by incrementally extending the set, initially consisting of these two subsequences, by searching for other subsequences which are non-overlapping and within the given range of the already found subsequences. ClusterMK on the other hand uses hierarchical clustering. Therefor, it considers the set of all subsequences in the time series and then repeatedly merges the top pair motif until its distance is not withing the given range anymore. The resulting set consists of subsequences, each representing a latent motif. The SetFinder algorithm calculates the distance between each pair of subsequences to find the subsequences with the most occurrences, i.e. set motifs.

Another state-of-the-art algorithm to find latent motifs is LearnMotif [4]. This methods discovers multiple latent learning motifs in a time series by maximizing an optimization problem on the frequency of similar subsequences. The objective function consist of a frequency part that counts how often the motifs occur in the time series and a violation part that penalizes similar latent learning motifs. However, the objective is not convex and thus it gives an approximate solution.

Most algorithms use the z-normalized Euclidean distance as the distance measure to gain robustness against horizontal stretching and displacements of subsequences in the time series.

The aforementioned methods were compared by Moczalla [10] using a synthetically generated benchmark. For latent motif discovery, the GrammarViz algorithm currently has the highest accuracy as well as the fastest execution time.

4.2 Matrix Profile

The matrix profile of a time series, as presented in [18], is a vector that for each subsequence contains the minimal z-normalized Euclidean distance to any other non-overlapping subsequence. Therefore, for each subsequence a so-called distance profile is computed, which contains the pairwise distance to each other non-overlapping subsequence. An entry in the matrix profile corresponds to the global minimum of the respective distance profile, i.e. the distance to the nearest neighbor subsequence.

The distance profile for a single subsequence can be calculated very fast using the MASS algorithm [12] which only needs $O(nlog(n))$ time, where n is the length of the time series, and thus is independent of the subsequence length. It exploits the fast Fourier transform to compute the sliding dot product of a subsequence with every other subsequence. Afterwards, the distance between two subsequences can easily be calculated. For mathematical details, we refer to the original work. The whole matrix profile can be computed even faster using the SCRIMP algorithm [19], which runs in $O(n^2)$ time. In this student project we will need the distance matrix of all distance profiles, an intermediate result of SCRIMP.

4.3 Exact maximum clique algorithms

The problem of finding the maximum clique in an arbitrary graph is NP-hard. This results from the fact that the clique decision problem, i.e. deciding whether a given graph G contains a clique of size k, is already NP-complete [6]. However, exact algorithms exist for sparse graphs.

As we expect that the distance graphs will be sparse, we give a short overview of exact algorithms that find maximum cliques in large sparse graphs. State-of-the-art methods for this problem are BBMCSP [16], PMC [15] and LMC [5]. In this work we focus on the latter two as their source code is publicly available.

The PMC algorithm uses a branch-and-bound strategy with efficient pruning. First, a heuristic is used to find a large clique. As the size of this clique is a lower bound for the maximum clique, all nodes are removed that can not be a part of a larger clique. At the same time, multiple upper bounds are used to stop as early as possible. For example, the number of colors in any greedy graph coloring is an upper bound for the size of the maximum clique. Although removed nodes only get marked at first, the graph periodically is recreated to speed up intersection operations in the search procedure. Additionally, the algorithm can be parallelized for roughly linear speedup.

LMC similarly preprocesses the graph by searching a large initial clique and removing as many nodes as possible. Afterwards it also employs a branch-and-bound strategy. For improvement of the coloring-based upper bound, the algorithm uses incremental MaxSAT reasoning. Therefore, the graph gets colored and implicitly converted into a partial MaxSAT instance which contains hard clauses that have to be satisfied and soft clauses that should be satisfied as far as possible. Then, if conflicting soft clauses are detected, a tighter upper bound can be derived.

5 Objectives

The goal of this research project is to develop a novel algorithm for latent motif discovery. Therefore, the main objectives are:

1. Use the Matrix Profile SCRIMP algorithm [19] for fast generation of a distance graph from the given time series. This unweighted graph contains a node for each subsequence. An edge between two nodes exists if and only if the corresponding subsequences are non-overlapping and $2r$-matching.

2. Analyze whether the resulting graphs are sparse for a fixed distance r, i.e. they contain only $O(n)$ edges where n is the number of vertices (subsequences). This is important in order to be able to choose an appropriate graph data structure and maximum clique algorithm. Alternatively, r might also be chosen based on lower percentiles of the distance matrix.

3. Run an exact maximum clique algorithm on that graph to find the top latent range motif. As we expect the graphs to be sparse for reasonably large distances r we can likely use an efficient algorithm for large sparse graphs, e.g. PMC [15] or LMC [5].

4. Evaluate the new approach using an existing benchmark [10]. It will be compared with other state-of-the-art methods [3, 4, 14, 17] regarding performance and runtime. The latter is particularly interesting as our presented algorithm is not approximative.

5. Visualize the motif discovery results, for example using Jupyter Notebooks.

This implementation will show whether the proposed approach is feasible for latent motif discovery. Furthermore, it will provide a basis for future improvements and experiments.

6 Methods and Results

The programming language used for the proposed pipeline will depend on the chosen algorithms and their available source code. For the matrix profile, multiple implementations in Python exist[2]. As the final matrix profile result only contains the index of the subsequence with minimal distance for each subsequence, not the pairwise distances (distance profile), the code may need to be modified to output those.

Because the main memory will not be able to hold the whole distance matrix, the distance graph should be built incrementally in a streaming fashion. Therefore, as soon as a distance is calculated the graph gets updated, so the full distance matrix does not have to be saved. A suitable sparse graph file format would be the Matrix Market Exchange Coordinate Format (.mtx)[3] as the following algorithms use it as a common input format. Also, it is supported by the Python SciPy library[4]. To find the maximum clique we either use existing implementations of the PMC[5] or LMC algorithm[6].

The benchmark we use for evaluation will consist of synthetic datasets from a previous master thesis [10]. We compare the presented algorithm against the state-of-the-art methods GrammarViz, ClusterMK, ScanMK, SetFinder, EMMA and LearnMotifs using point-based precision, recall and F_1-score as used in [10]. We also measure the runtime of every algorithm for each dataset.

Finally, the project code will be uploaded to a public GitHub repository in order for it to be available for further experiments.

[2] https://github.com/zpzim/SCAMP, https://github.com/TDAmeritrade/stumpy, https://github.com/matrix-profile-foundation/matrixprofile

[3] https://math.nist.gov/MatrixMarket/formats.html#MMformat

[4] https://docs.scipy.org/doc/scipy/reference/io.html

[5] https://github.com/ryanrossi/pmc

[6] https://home.mis.u-picardie.fr/~cli/EnglishPage.html

References

[1] H. Abe and T. Yamaguchi. Implementing an Integrated Time-Series Data Mining Environment - A Case Study of Medical KDD on Chronic Hepatitis. In *1st international conference on complex medical engineering (CME2005)*, 2005.

[2] I. P. Androulakis, J. Vitolo, and C. Roth. Selecting maximally informative genes to enable temporal expression profiling analysis. *Proc. of Foundations of Systems Biology in Engineering*, page 23, 2005.

[3] Anthony Bagnall, Jon Hills, and Jason Lines. Finding motif sets in time series. *arXiv preprint arXiv:1407.3685*, 2014.

[4] Josif Grabocka, Nicolas Schilling, and Lars Schmidt-Thieme. Latent Time-Series Motifs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(1):1–20, 2016.

[5] Hua Jiang, Chu-Min Li, and Felip Manyà. Combining Efficient Preprocessing and Incremental MaxSAT Reasoning for MaxClique in Large Graphs. In *Proceedings of the twenty-second European conference on artificial intelligence*, pages 939–947, 2016.

[6] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[7] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Pranav Patel. Finding Motifs in Time Series. In *Proc. of the 2nd Workshop on Temporal Data Mining*, pages 53–68, 2002.

[8] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144, 2007.

[9] Amy McGovern, Derek H. Rosendahl, Adrianna Kruger, Meredith G. Beaton, Rodger A. Brown, and Kelvin K. Droegemeier. Anticipating the formation of tornadoes through data mining. 2007.

[10] Rafael Moczalla. A Synthetic Motif Generator. Master's thesis, Humboldt University of Berlin, 2020.

[11] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. Exact Discovery of Time Series Motifs. In *Proceedings of the 2009 SIAM international conference on data mining*, pages 473–484. SIAM, 2009.

[12] Abdullah Mueen, Yan Zhu, Michael Yeh, Kaveh Kamgar, Krishnamurthy Viswanathan, Chetan Gupta, and Eamonn Keogh. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance, August 2017. `http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html`.

[13] Tim Oates, Matthew D. Schmill, and Paul R. Cohen. A Method for Clustering the Experiences of a Mobile Robot that Accords with Human Judgments. In *AAAI/IAAI*, pages 846–851, 2000.

[14] Pranav Patel, Eamonn Keogh, Jessica Lin, and Stefano Lonardi. Mining Motifs in Massive Time Series Databases. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 370–377. IEEE, 2002.

[15] Ryan A. Rossi, David F. Gleich, Assefaw H. Gebremedhin, and Md. Mostofa Ali Patwary. Parallel Maximum Clique Algorithms with Applications to Network Analysis and Storage. *arXiv preprint arXiv:1302.6256*, 2013.

[16] Pablo San Segundo, Alvaro Lopez, and Panos M Pardalos. A new exact maximum clique algorithm for large and massive sparse graphs. *Computers & Operations Research*, 66:81–94, 2016.

[17] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P. Boedihardjo, Crystal Chen, and Susan Frankenstein. GrammarViz 3.0: Interactive Discovery of Variable-Length Time Series Patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(1):1–28, 2018.

[18] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. Ieee, 2016.

[19] Yan Zhu, Chin-Chia Michael Yeh, Zachary Zimmerman, Kaveh Kamgar, and Eamonn Keogh. Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 837–846. IEEE, 2018.