Adjusting Monitored Experiments to Real-World Cases by Matching Labeled Time Series Motifs

Christian Moewes and Rudolf Kruse

School of Computer Science, Otto-von-Guericke University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
Tel: +49 (391) 67 ext. {11358, 18706}
Fax: +49 (391) 67 12018
E-Mail: {cmoewes, kruse}@iws.cs.uni-magdeburg.de

Abstract

In this paper we devote ourselves to the difficulty of fitting human designed experiments to real-world cases. We decompose this problem into two smaller subproblems: 1.) The search of recurrent patterns in temporal sequences, so called motifs that are deemed to be discovered in both the experiments and the real observations and 2.) the matching of motifs to linguistic terms which are possibly available as domain knowledge. Therefore we describe an effective time series representation that enormously speeds up the search for these motifs. We present some approaches to adjust the designed experiments with the help of the discovered motifs. Finally, we conclude our work and give prospects to possible extensions.

Keywords: Multivariate Time Series Analysis, Motif Discovery, Labeling, Frequent Pattern Mining.

1 Introduction

Conducting field tests of complex systems to evaluate their behavior is usually expensive and time consuming. One requirement is that the designed tests should be as similar as the behavior of their pendants which are produced in series and used in the real world. Based on these experiments which quantitatively describe criteria (e.g., lifetime, errors, loadings), the quality of a system might be improved. For instance sources of error can be found and remedied in the next generation of systems.

In order to evaluate these criteria, sensor data are recorded over long periods of time from the test and the real objects, respectively. Existing data analysis methods are based on one-dimensional load spectra which are utilized to compare the tests to the reality. These methods are completely time-independent. However, the timely behavior of real-world systems might contain many different processes that have not been considered in the tests so far. Additionally, the tests may contain procedures that do not meet reality at all.

Data mining methods that consider time as additional variable are discussed to some extend in Section 2. By means of these methods to analyze time series we are able to discover interesting recurrent patterns, so-called *motifs*. We describe a very effective algorithm to find such motifs in Section 3.

One task then would be to match discovered motifs to test criteria of the time series. Every time series containing a motif thus can be labeled by at least one test criterion if the relevant test has been designed thoroughly. Hence a motif and its label can be regarded as a discovered rule's antecedent and consequent, respectively. In Section 4 we explain how motifs are labeled by means of expert's knowledge.

Having identified a set of rules in all experiments, we try to retrieve a subset of it in real-world data. As a consequence unseen time series can be assigned and compared to the given experimental criteria. This part of our work is specified in Section 5.

Thence it is possible to adjust the experiments to the given real-world cases as follows. For example if the motifs of an unseen time series cannot be found in any experiment, then the tests should be adjusted such that every new motif occurs at least once. Then again motifs that only exist in test data and not in the real-world should be removed since they do not seem to be relevant. Section 6 clarifies further details of this approach. We finally summarize our work and give prospects to open questions and problems in Section 7.

2 Time Series Data Mining

In research and development, data mining in time series has gained incredibly big attention in the last years. Meanwhile time series are simply ubiquitous in areas such as finance, medicine, biometry, chemistry, astronomy, robotics, networks and industry. So-called time series databases save an additional time stamp for every stored dataset. A time series can be arbitrarily long and potentially involve several dimensions (attributes, channels, sensors). Then a time series is no longer called univariate but multivariate.

A big challenge is the search for useful information in time series. Today we distinguish between the following time series data mining tasks: Clustering [1], classification [2], motif discovery [3], rule generation [4], visualization and anomaly detection [5].

Due to the plenty of data, many of those problems can usually be broken down to the search for recurrent sequences in the time series that are similar to each other. In order to find these sequences one has to define a similarity measure that compares two sequences. Most publications in that field use the Euclidean distance

$$d(Q,C) = \sqrt{\sum_{i=1}^{w}(q_i - c_i)^2} \tag{1}$$

between two normally distributed sequences $Q = (q_1, \ldots, q_w)^T$ and $C = (c_1, \ldots, c_w)^T$ of length w as basis for the similarity measure. If we use (1) to measure the similarity, usually lots of comparisons must be performed to find some motifs. Moreover, the capacity of every fast main memory is most of the time to small to load all original data at once.

2.1 Memory-Efficient Representations

Owing to these many and especially slow accesses to the original data on the hard disc, one should us an approximation of every time series that fits into the main memory of a computer and contains all essential and interesting features. There are dozens of different kinds of time series approximations, e.g., discrete Fourier transform (DFT), discrete wavelet transform (DWT), piecewise linear models (PAA), piecewise constant models

(APCA), singular value decomposition (SVD), symbolic representations. The latter ones benefit by being applicable to algorithms that originate from text processing and bioinformatics, e.g., hashing, Markov models, suffix trees, etc.

In current research the symbolic representation of Lin and Keogh [6] wins out even over well-known approximations. Their symbolic aggregate approximation (SAX) transforms a univariate time series sequence into a word of defined length n over a chosen alphabet A with $|A| = a$. The SAX algorithm is rather simple but intuitive.

Firstly, the sequence is separated into n equal parts. Then the mean of every interval is computed as representative of all values in that interval. This method is also called piecewise aggregate approximation (PAA) [7].

After that step the essentially shorter sequence of mean values is discretized as follows. Every mean value of the PAA sequence is assigned to one of the a letters such that the occurrence of every letter in the sequence is equally probable. This is achieved by assuming that the PAA sequence's range of values is normally distributed. Furthermore, this distribution is split up into parts such that all parts share the area under the Gaussian curve. This assumption can be made due to the following fact. Long time series may not be normally distributed, but their short sequences certainly are to a high degree [6].

While other symbolic representations generate a word from time series data as well, SAX is yet one of a kind compared with them. It does not only compress the sequence. SAX also enables us to measure a distance $d^*(Q, C)$ between two SAX words which is a lower bound of the Euclidean distance (1) between the original sequences Q and C, formally

$$d^*(Q, C) \leq d(Q, C).$$

For the rest of the paper we assume that the similarity is determined by the Euclidean distance (1). So, a lower bound means that if two SAX words are dissimilar, then their original sequences are dissimilar as well. Consequently, algorithms that are based on SAX produce identical results compared to algorithms that work with the original data. Merely similar SAX words should be compared in the Euclidean space again. Fortunately, those accesses to the original data are only very rare since most of the comparisons are based on dissimilar sequences.

Having a memory-efficient representation we can concentrate ourselves on finding similar sequences efficiently. In the following we proceed from the assumption that every time series is approximated by SAX since the next algorithms are based on hashing.

3 Motif Discovery in Time Series

If we are able to find recurrent sequences that are similar to each other, then problems such as clustering or classification of time series are much easier to solve. These similar sequences are called *motifs* due to the vocabulary that is used in bioinformatics. This originates from the fact that in this domain, motifs correspond to recurrent strings (usually from a DNA).

In the article from Chiu et al. [8] SAX is associated with motif discovery in univariate time series for the first time. In order to find all motifs of a time series of length l, it is separated by a sliding window with certain width w into $(l - w + 1)$ sequences. Every sequence is transformed into a SAX word and saved into a $(l - w + 1) \times n$ matrix which we call SAX matrix.

The positions of possible motifs are then guessed using the random projection algorithm proposed by Buhler and Tompa [9]. Actually, the positions are found by pairwise comparisons of the SAX words. So, for each of those $(l - w + 1)^2$ comparisons, we firstly reserve one entry in a collision matrix $\mathbf{M}$ which can be implemented efficiently by a hash table. In the beginning, let every entry $\mathbf{M}(i, j)$ be zero for $1 \leq i, j \leq l - w + 1$.

Although usually $n << w$, it is not preferable to compare every single character of the saved SAX words in the matrix with each other. Buhler and Tompa rather had the idea that there exist so-called *don't care symbols* of which we do not know where they might be in the words. These symbols would correspond to, e.g., a noisy motif, a dilation/contraction of a temporal sequence.

Accordingly the SAX matrix is projected down to $1 \leq k < n$ randomly chosen columns. Afterwards all rows of the projected matrix are compared with each other. If two projected SAX words in the rows i and j are equal, then the value in $\mathbf{M}(i, j)$ is incremented by one.

The projection is repeated t times since one can assume that some of the hidden motifs will share one entry in $\mathbf{M}$ after t iterations. Additionally, it is improbable that many random sequences will collide together with an already found motif. Therefore they would have to be identical to this motif in all k positions.

Since the algorithm cannot know if a collision entry in $\mathbf{M}$ is a motif or not, the user must specify a threshold $1 \leq s \leq k$. All $\mathbf{M}(i, j) \geq s$ thus would be motif candidates. Remembering that we deal with temporal and not DNA sequences, the problem of motif discovery becomes harder as we find similar occurrences of the i-th sequence in its direct neighborhood. Those sequences which are named *trivial matches* [8] are heuristically removed from the set of potential motifs at the end of the discovery.

Although comparatively, many parameters have to be determined, i.e., n, a, w, k, t, s, random projection is robust against slight changes of the SAX parameters n and a as well as the projection size k [8]. Also the number of projections t can be set large enough in order to create some collisions. However, there are two questions left: How many and in particularly what kind of motifs do we have to find?

If we set w and s too large on the one hand, then we may not find lots of “short” motifs. On the other hand, we will get completely different results if we set w and s too small. Then we will probably find many random consensuses that do not correspond to any real motif. Therefore, the choice of these two parameters should be made carefully. Expert's knowledge may help in such a situation.

As a side remark, we would like to mention that Yankov et al. [10] extended time series random projection to a non-Euclidean distance measure, i.e., *uniform scaling*. With this method one can find motifs that are not exactly w time stamps long. This approach is indeed limited such that w must be chosen based on the respective application.

3.1 Subdimensional Motifs

The random projection to find time series motifs [8] was originally only designed for one-dimensional datasets. If we deal with multivariate time series, then there exist several ways how to tackle this problem.

The simplest idea is to map the p dimensions down onto one and then use random projection. For instance, Tanaka et al. [11] have transformed the input dimensions by means of principal component analysis (PCA) into solely the first principal component. Finally, the approach from Chiu et al. [8] could be applied to the new univariate time series.

A first approach of Minnen et al. [12] is founded on the idea that p dimensions also generate p SAX words. These SAX words are then concatenated and treated like a SAX representation of a long univariate time series. As a consequence, the method from Chiu et al. [8] can be applied in this case as well.

Though notice that both approaches can only discover motifs that span all dimensions. This will be problematic in particular if we *a priori* do not know in which of the dimensions we can observe any motif. In practice it can also happen that a time series motif's attributes can differ quite from another one's attributes. Such multivariate time series motifs that do not span all dimensions are called *subdimensional*.

Formally, we denote a multivariate sequence as $w \times p$ matrix which stores w real values for each of the p attributes. We define the distance d_{mult} of two multivariate sequences $\mathbf{Q} = [Q_1, \ldots, Q_p]$ and $\mathbf{C} = [C_1, \ldots, C_p]$ by the Euclidean norm

$$d_{\text{mult}}(\mathbf{Q}, \mathbf{C}) = \|\mathbf{d}\|_2 = \sqrt{\sum_{j=1}^{p} |d_j|}$$

whereas $\mathbf{d} = (d_1, \ldots, d_p)$ and $d_j \equiv d(Q_j, C_j)$ corresponds to the Euclidean distance (1) between Q_j and C_j for $1 \leq j \leq p$.

According to our literature research, there is so far only one approach that tries to discover subdimensional motifs. Minnen et al. [13] improve their original idea to concatenate the SAX words of every dimension. They increment the collision matrix $\mathbf{M}$ *per attribute* at the appropriate entry for every projected SAX word that matches another one.

Afterwards all elements of $\mathbf{M}$ that are greater than s are picked out and must be examined further. Note that although we have two positions for each pair of sequences, nonetheless we do not know its relevant dimensions. Furthermore, there is not any assignment of the pairs of sequences to the potential motifs yet.

Before we can perform this assignment, we have to extract the subdimensions of the sequences by means of the following naïve idea. For every pair of sequences we sort all distances $d_1, \ldots, d_p$ in an ascending order. Then the distance is accumulated in that order for every single dimension until a certain threshold $r_{\max}$ is exceeded. The attributes of the smallest distances thus correspond to the pair of sequences' relevant subdimensions.

These heuristics can be also improved by not regarding attributes having smallest distances, but using only probably relevant attributes to compute the distance [13]. Therefore one estimates the empirical frequency distribution $P(d_j)$ over the distances between some non-trivial matches for every dimension $1 \leq j \leq p$ by random sampling. Later on the distances $d_1^*, \ldots, d_p^*$ are computed for every entry $\mathbf{M}(i, j) \geq s$. If the value of the cumulative distribution function $P(d_j \leq d_j^*)$ is smaller than the dimension relevance r_{rel} which is specified by the user, then the j-th dimension be relevant.

Determined all pairs of subdimensional sequences, the trivial matches have to be eliminated as it was done in the univariate case of motif discovery. With this idea [13], motifs do not need to span all dimensions. This would an asset compared to [11, 12] when the set of attributes does contain, e.g., very noisy signals, uninformative dimensions.

Disadvantages of this method for subdimensional motif discovery are the threshold parameter $r_{\max}$ and r_{rel}, respectively. Both extremely depend on the sequence length w. So, if domain knowledge is present, then it is suitable to use $r_{\max}$ as threshold. Otherwise one must estimate the distribution $P(d)$ and handle with r_{rel}.

4 Labeling Discovered Motifs

Having identified a set of subdimensional motifs, we merely found multivariate time series sequences of certain length w that recur at least twice. Note that we can find random motifs accidentally as well. Thus it is probable that a motif which recurs only twice might not be what we are looking for.

Yet, motifs that recur more often should be labeled meaningfully from mainly experts who designed the experiments. They usually possess the necessary knowledge to interpret both simple and complex curve progressions. This labeling can be done, e.g., by means of the test criteria.

If there is no expert's knowledge available, then one can fall back on methods from fuzzy set theory (FST) [14]. In FST one tries to model imprecise, vague or even uncertain concepts, e.g., sensor measurements, such that the human being obtains a better understanding of these concepts.

For instances, every attribute can be regarded as linguistic variable [15]. In doing so, the attribute's range of values is separated into a so-called fuzzy partition. Every partition is described by a fuzzy set A. Thus every value x can be assigned to a membership degree $\mu_A(x) \in [0, 1]$ of the fuzzy set A.

We consider the measured velocity v as an illustrating example. The velocity can be described by some linguistic terms, e.g., *fast*, *medium*, *slow*. Every expression corresponds to a fuzzy partition which then again is described by a fuzzy set, i.e., A_{fast}, A_{medium}, A_{slow}.

If we want to assign a discovered motif to a linguistic term, for example we can compute the mean $\bar{v}$ of all velocity values in the respective sequence. The linguistic term with the highest of the three membership degrees $\mu_{A_{\text{fast}}}(\bar{v})$, $\mu_{A_{\text{medium}}}(\bar{v})$ and $\mu_{A_{\text{slow}}}(\bar{v})$ is then labeled to the motif.

If the experiments are designed thoroughly (i.e., they do not contain contradictory linguistic terms), then it is assumed that a time series which contains the labeled motif can be labeled in the same way. If this is not the case, we can firstly compute the relative frequencies of labeled motifs in a time series, and secondly assign several labels to this time series to a certain degree.

Every labeled motif and its linguistic term can thus represent the antecedent and the consequent of a rule, respectively. We can further hope that such a consequent corresponds to a test criterion. From the monitored experiments we finally obtain a set of rules which can be interpreted in terms of natural language by more or less great efforts.

5 Matching Labeled Motifs

So far we solely considered the data coming from the field tests. The assumption in Section 1 was that these trials are designed and performed very thoroughly. The system that needs to be tested may behave completely different in a real-world environment, e.g., when it is utilized by an end user. In this situation we face the problem that systems under real-world loads might not follow any designed schedule model.

Usually the only thing what remains to evaluate these systems is monitored sensor data that hopefully contains motifs similar to the ones from the experiments. These real-world data is foremost approximated memory-efficiently (see Section 2) before we try to find motifs in the data (cf. Section 3). Now we can try to label the newly discovered motifs with similar linguistic terms by means of the already labeled motifs from the field tests. In machine learning, this would correspond to classification that is based on unsupervised learning.

Remember that it is very important to choose an adequate distance measure in order to compare two motifs. For example Lin und Keogh [6] have developed not only SAX but the so-called MINDIST function which computes the distance between two SAX words. It is preferable to use this function since the sequences are stored as SAX words anyway. Of course, other distance measures, e.g. (1), could be used as well.

No matter which measures we choose, eventually every real-world time series can be matched with previously unknown criteria. Taking everything into account, we can state that a classification into different criteria is thus a trivial consequence. Nevertheless, we have to consider that this classification should be carried out rather fuzzy than crisp. Accordingly, the usage of fuzzy clustering methods [16] seems to be desirable.

6 Adjusting the Experiments

Having finally discovered all motifs of the real-world data and labeled them to the already existing ones, experts should have a closer look at the results of the matching. The goal should be to adjust the original experiments such that they will resemble the time series more than before.

In total, three different possibilities have to be distinguished. If an unseen motif (coming from any real-world case) could be matched easily with a motif from a field test, then we can assume that we found some important feature of the system behavior. At any rate, such characteristics should be kept in all experiments in the next generation of system tests.

Experts would probably react differently in the case that a motif is exclusively discovered in field tests and not in real-world case. Such a feature should most likely be removed from the experiments after expert opinion. It is clear that this type of motif does not matter at all.

If there are in turn motifs in unseen time series that do manifest themselves in any trial, then experts have to adjust at least one trial. After all, this motif seems to be a recurrent feature of the system which occurred either never or not often enough in the field tests.

When all motifs are examined and the test design is improved, the next generation of experiments can be performed. The gained knowledge about, e.g., loading, service live, which results from the tests should consequently be more consistent with the serial product used in the reality. Finally, these experiences can possibly provoke enhancements of the systems.

7 Conclusions and Future Work

In this paper, we dealt with the question how field tests of systems that are produced in series might be adjusted to real circumstances. We especially pursued the efficient analysis of multivariate time series. This is due to the fact that in practice there is usually nothing but monitored data from many sensors available.

We argue that symbolic representations (in particular SAX) are comparatively superior in the data analysis of time series. Furthermore we can find very efficient methods which allow us to find recurrent sequences, i.e., motifs, in multivariate time series very fast. Unfortunately, there is not any satisfying heuristics to handle the vast number of parameters which influence the search and thus the success of the application.

Some unseen time series can be matched with the field trials by means of linguistic terms and discovered motifs in both the tests and real-world cases. The linguistic terms may either result from human experts or be generated from temporal sequences. After the clustering of all time series, the same experts might adjust the original tests to the observed reality.

In our work, we did not cover the discovery of errors and anomalies [5] in the measured data. We are rather concerned how to support the optimization of test procedures. While we are looking for frequent patterns, anomalies and errors usually occur highly infrequent.

The idea was born during an external funded project that we are working on with an industrial partner. This partner supplies us with much data from experiments and matched test criteria which shall describe the experiments. Usually in the industry one-dimensional load spectra are used to compare experiments with the reality. Our work is meant as addition to existing methods for data analysis.

So far we proceeded in our efforts to the discovery of some meaningful motifs. Though given the nondisclosure of the project, we are not allowed to publish either a name, or results, or some visualization of the data.

The next step will be the application of fuzzy clustering methods to all found subdimensional motifs in order to generate rules for some of the time series. This should help us to analyze unseen time series. Moreover, we plan both to automate the labeling of motifs with the linguistic terms and to verify them with the given test criteria. In order to do so, we will not only restrict ourselves to local trends in sequences (e.g., mean values) but we will consider the variability and the length of a motif as well.

The latter measure will probably be hard to obtain since the length of any motif is unknown in principle. If we can develop an algorithm that discovers motifs of different lengths, then we could focus on the next challenge in this field. Approaches from [17, 10] may be helpful to solve this task.

References

[1] Lin, J.; Vlachos, M.; Keogh, E. J.; Gunopulos, D.: Iterative Incremental Clustering of Time Series. In: *EDBT* (Bertino, E.; Christodoulakis, S.; Plexousakis, D.; Christophides, V.; Koubarakis, M.; Böhm, K.; Ferrari, E., Hg.), Bd. 2992 von *Lecture Notes in Computer Science*, S. 106–122. Springer. ISBN 3-540-21200-0. 2004.

[2] Xi, X.; Keogh, E. J.; Shelton, C. R.; Wei, L.; Ratanamahatana, C. A.: Fast time series classification using numerosity reduction. In: *ICML* (Cohen, W. W.; Moore, A., Hg.), Bd. 148 von *ACM International Conference Proceeding Series*, S. 1033–1040. ACM. ISBN 1-59593-383-2. 2006.

[3] Patel, P.; Keogh, E. J.; Lin, J.; Lonardi, S.: Mining Motifs in Massive Time Series Databases. In: *ICDM* [18], S. 370–377. 2002.

[4] Höppner, F.: Discovery of Temporal Patterns: Learning Rules about the Qualitative Behaviour of Time Series. In: *PKDD* (De Raedt, L.; Siebes, A., Hg.), Bd. 2168 von *Lecture Notes in Computer Science*, S. 192–203. Springer. ISBN 3-540-42534-9. 2001.

[5] Lin, J.; Keogh, E. J.; Lonardi, S.: Visualizing and Discovering Non-Trivial Patterns In Large Time Series Databases. *Information Visualization* 4 (2005) 2, S. 61–82.

[6] Lin, J.; Keogh, E. J.; Lonardi, S.; Chiu, B. Y.: A symbolic representation of time series, with implications for streaming algorithms. In: *DMKD* (Zaki, M. J.; Aggarwal, C. C., Hg.), S. 2–11. ACM. 2003.

[7] Keogh, E. J.; Chakrabarti, K.; Pazzani, M. J.; Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems* 3 (2001) 3, S. 263–286.

[8] Chiu, B. Y.; Keogh, E. J.; Lonardi, S.: Probabilistic discovery of time series motifs. In: *KDD* (Getoor, L.; Senator, T. E.; Domingos, P.; Faloutsos, C., Hg.), S. 493–498. ACM. ISBN 1-58113-737-0. 2003.

[9] Buhler, J.; Tompa, M.: Finding Motifs Using Random Projection. *Journal of Computational Biology* 9 (2002) 2, S. 225–242.

[10] Yankov, D.; Keogh, E. J.; Medina, J.; Chiu, B. Y.; Zordan, V. B.: Detecting time series motifs under uniform scaling. In: *KDD* (Berkhin, P.; Caruana, R.; Wu, X., Hg.), S. 844–853. ACM. ISBN 978-1-59593-609-7. 2007.

[11] Tanaka, Y.; Iwamoto, K.; Uehara, K.: Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle. *Machine Learning* 58 (2005) 2-3, S. 269–300.

[12] Minnen, D.; Starner, T.; Essa, I. A.; Isbell, Jr, C. L.: Improving Activity Discovery with Automatic Neighborhood Estimation. In: *IJCAI* (Veloso, M. M., Hg.), S. 2814–2819. 2007.

[13] Minnen, D.; Isbell, Jr, C. L.; Essa, I. A.; Starner, T.: Detecting Subdimensional Motifs: An Efficient Algorithm for Generalized Multivariate Pattern Discovery. In: *ICDM*, S. 601–606. IEEE Computer Society. 2007.

[14] Dubois, D.; Prade, H. (Hg.): *Fundamentals of Fuzzy Sets*. Boston, MA, USA: Kluwer Academic Publishers. ISBN 978-0-792-37732-0. 2000.

[15] Zadeh, L. A.: The Concept of a Linguistic Variable and its Applications to Approximate Reasoning–I. *Information Sciences* 8 (1975) 3, S. 199–249.

[16] Höppner, F.; Klawonn, F.; Kruse, R.; Runkler, T.: *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. New York, NY, USA: John Wiley & Sons Ltd. ISBN 978-0-471-98864-9. 1999.

[17] Oates, T.: PERUSE: An Unsupervised Algorithm for Finding Recurrig Patterns in Time Series. In: *ICDM* [18], S. 330–337. 2002.

[18] *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), Maebashi City, Japan, December 9-12, 2002*. IEEE Computer Society. ISBN 0-7695-1754-4. 2002.