Robust Multivariate Pattern Mining with Inaccurate User Annotation

Victoria Citiriga, Softing ROM s.r.l.
Dr. Albert Krohn, 21data.io
Dr. Christopher Anhalt, Softing Industrial Automation GmbH

Correct identification of states or regimes in a process is essential to a high variety of data mining tasks. In anomaly detection, the main focus is abnormal values that appear in specific contexts meaning that a prior isolation of similar stages of the data is required.
In the current study, we look at manufacturing processes monitored by multivariate time series. The segmentation of the series is based on multiple user annotations corresponding to a particular event occurring multiple times. Further, there is no additional information about the importance of the variables captured. The solution proposed will align the annotations based on maximizing the cross-correlation between them and selecting the variables that are most representative. Moreover, it allows to identify and ignore incorrect annotations inadvertently inserted by the user. It also supports negative annotations – samples of the data that might be mistaken for the event of interest.

1 Introduction

Anomaly detection represents the identification of abnormal values in a specific context. It has gained an increased importance in the last decade due to its various applications in domains such as fraud detection, sensor networks, cybersecurity, industrial damage detection. The increase in data volumes and velocity makes it impossible to identify all anomalies visually and gave rise to the need to automate the process.

The success of any anomaly detection system depends upon the correct definition and isolation of what a normal behavior is.

In our proposal we approach this in a supervised way by letting expert users annotate normal areas in the time series by simple interaction with time series plots.

In practice, manual annotations made by experts are not perfectly aligned nor of the same length (e.g.: some might define the production state of a system as starting when the process is at full speed, while others might consider the ramp up of a system as part of the same state), which can make identifying the characteristics and the variables that define the state harder.

1.1 Use case and motivation

The current study looks at multivariate time series data sets in which anomaly detection is to be applied. As a data preparation step, user annotations are used in order to segment the data or identify patterns which are the basis for further processing. The user marks areas of interest and that information can later be used as input to supervised machine learning algorithms. It is expected that the user annotates multiple examples of the same patterns occurring in the time series to increase the number of training inputs to the following machine learning algorithms. The current algorithm improves the user annotations to increase robustness for following data processing steps. In particular, the steps undertaken are to:

- adjust annotations made by experts (by shifting, adjusting length) in such a way that they look alike and are aligned best
- identify the variables that describe them best
- identify annotations that are wrong (do not exhibit the same behavior as the others) and discard them as outliers from the annotations set
- take into account so called negative annotations - this is a pattern that is similar to the real pattern, and can be confused with it. For example, let's assume that the goal is to mine for the production phase of a process. This production phase can be identified by the behaviour of some variables. Let's assume the idle phase of the same process looks the same as the production phase for one variable X that's not included in the initial set of variables. By specifying a negative annotation representing the idle phase we are helping the algorithm know the difference between the two phases and

that it should exclude variable X when running.

These contributions constitute a vital preprocessing step for pattern mining, since the results of any pattern miner are highly dependent on the accuracy of the time series representations fed in and the relevance of the variables that are taken into consideration when building the pattern miner.

2 Robust pattern mining

2.1 related work

Pattern miners consist of two components: a time series representation and a distance measure.

One interesting contributing in the area of pattern mining is the Piecewise Aggregate Approximation (PAA) approach by Keogh et al (Keogh et al. n.d.) By creating segments of equal length and considering the mean of such segments, the approach has the advantage of dimensionality reduction and shows promising results even when compared with more sophisticated transformations.

Symbolic Aggregate Approximation or SAX introduced by (Lin et al. 2007) can be considered an extension of the PAA approach. First, it normalizes the data, splits it into equal sized segments and approximates each of them with the mean. Afterwards, it splits the distribution space into a number of equiprobable segments equal to the alphabet size, and assigns a symbol to each one.

(Gao & Lin 2018) introduced an algorithm called Hierarchical-Based motif enumeration (HIME) that detects variable length motifs. The starting point of their algorithm is a modified SAX miner, which uses numerosity reduction to speed the algorithm up (neighbouring subsequences are ignored if the PAA distance between them is less than a threshold).

Most common distance measure used are euclidean and dynamic time warping (Berndt D.J.,Clifford J. ,1994). The former measure has the drawback of carrying quadratic complexity on the length of the sequences involved.

Although the described algorithms show promising results, they are mostly designed for the univariate case. When faced with multiple variables, a proper variable selection is necessary followed by an aggregation method.

The current study comes as a solution for the variable selection/ranking problem and for the motif cleaning.

2.2 Algorithm for robust pattern mining

Consider a set of k variables where $Y_{t,k}$ = value of variable k at time t, and $\boldsymbol{Y_k}$ is the vector of all samples of variable k. Also consider a set of j annotations (or motifs) that represent occurrences of the same event.

Also consider ***Anno j*** as the annotation defined by t_{j1} (start time) and t_{j2} (end time) .

We define $\boldsymbol{Y_{k,Anno\ j}}$ = equidistant time series values taken by variable k between times t_{j1} and t_{j2} and the sampling rate as the difference (in seconds) between two consecutive timestamps. The length of the variable vector will be:

$$Len\ \boldsymbol{Y_{k,Anno\ j}}=(t_{j2}-t_{j1})\cdot SamplingRate$$

The proposed procedure has the following parameters:

- *MaxLag*: the maximum number of time points by which an annotation or motif will be shifted to the left/right[1]
- *MinCorr* : the minimum correlation value that a variable has to have in order to be considered significant in describing a pair of motifs ; the default value is the 75th percentile of the maximum value of correlations for each variable across all lags
- *NoVars*: the minimum number of significant variables that should describe the relationship between each motif and the main motif; any motif that has less than NoVars significant variables to describe its relation to the main motif is considered as an outlier motif

Apart from the parameters described, the annotations that need to be adjusted and the set of variables, one optional input consists of *negative annotations* or motifs.

The current method will consist of 3 main steps, described in detail below. The first step consists of identifying the main motif (the motif that correlates best with all others), the second step identifies variables that should be ignored and the third step aligns all motifs to best match the main motif and identifies possible outlier motifs (motifs that aren't correlated with the main motif).

Step 1: Main motif identification

Pattern miners normally rely on having *one* representation of a pattern. We are using the

[1] the shifting of the user annotation is done to improve the start and end times of user annotation in order to get the most powerful pattern miner

multiple annotations of the same event in time given by the user to identify the *best* or *main motif* out of all annotations for the same event. This motif will be the one loaded into the pattern miner. The main motif should have the best representative power for the event to mine for.
The identification of the main motif is crucial because all other motifs will be shifted to maximize their correlation with this main motif. Also, possible negative motifs or annotations will be compared with this main motif and based on this comparison the highly correlated variables will be considered misleading and thus ignored.
Considering a MaxLag (ML) set by a user, a correlation matrix for each 2 pairs of motifs will be computed (for a number of j annotations, the total number of matrices will be permutations of j taken by 2 since the length of the motifs might differ and the second annotation in the pair will be cut/extended to match the length of the first one).
For example, considering ***Anno i*** defined on $[t_{i1}, t_{i2}]$ and ***Anno j*** defined on $[t_{j1}, t_{j2}]$, the result will be a 2ML·k correlation matrix for the 2 annotations, as shown in the appendix.

The defined correlation matrix keeps the first motif as fixed, and shifts the values for the second one to the left and right (this means that the second motif will be adjusted - extended/cut - to match the length of the first motif). The notation for the second motif (e.g.: $\boldsymbol{Y}_{[t_{j1-ML}, t_{j1-ML+Len\ Anno\ i}]}$) just makes sure that we are comparing variables of the same length.
To identify the main motif or annotation, an aggregate measure of the correlation of each motif with all other motifs is needed.
However, for each motif there exists one correlation matrix for each other motif we are comparing it to.
The goal is to find a way to quantify the strength of the relationship between one motif and all other motifs through a single number. To achieve this, the approach is the following:

- for each two motifs, based on the correlation matrix defined in the appendix, the maximum correlation for each variable across all possible shifts is computed
- not all variables will be significant in describing that pair of motifs, so when aggregating these values, a measure that is robust to extreme values will be chosen (e.g.: a percentile)
- the 75th percentile of the previously obtained values is considered, and this will provide a measure of how well that pair of motifs correlates overall
- for a single motif, there are j-1 (where j is the number of annotations/motifs) such pair-correlation-values that need to be aggregated once more
- the 75th percentile of these values is computed, obtaining one value per motif which provides a measure of how good this motif correlates with all other motifs under all possible time shifts
- the motif that has the highest value will be the main motif

Schematically, the process is described in the appendix.
The algorithm is summarized below:

Alg1: Main motif identification
Input: k variables, n motifs and user set parameters: MaxLag, NoVars Output: Main motif For each 2 motifs i and j out of the n motifs: Calculate Corr(***Anno i***, ***Anno j***) for all vars and possible lags For each variable v: take the maximum of all Corr(***Anno i***, ***Anno j***) across lags as $MaxCorr_{ij,v}$ Calculate $MaxCorr_i$ = 75^{th} percentile for each motif i out of the $MaxCorr_{ij,v}$ (j=1..n, j ≠ i,v=1..k)
Return: *MainAnno =Anno i* for which: $MaxCorr_{MainAnno}$= max($MaxCorr_i$)

Step 2: Identify variables that should be ignored

For each of the negative motifs/annotations we will define correlation matrices as shown by (1) with the main motif and find those variables that at any lag, have a correlation of at least MinCorr.

Alg2: Ignored variables
Input: k variables, m negative annotations/motifs, Main motif, and set parameters: MaxLag Output: Ignored variables For each negative annotation j: Calculate NegCorr=Corr ($MainAnno, Anno\ j$) $IgnoredVars = \underset{var=1..k}{\text{argwhere}}\ Corr(NegativeAnno\ i, MainAnno) > MinCorr$ ***IgnoredVars***=all var v for which NegCorr>MinCorr
Return: ***IgnoredVars***

Step 3: Identify best lag and best variables that describe all motifs

In order to identify the best shift and best variables that describe the motifs, an iterative procedure is implemented.
For each motif i, the lag and variable for which we get the highest correlation is chosen:

$$Lag_i, Var_i = \text{argmax}_{lag=-ML,ML,var=1,k} Corr(Anno\ i, MainAnno) \quad (2)$$

After deciding the first lag and first variable, we proceed to look at all other variables except the ones in ***IgnoredVars*** that have (for that same lag) a correlation of at least MinCorr.
If there are at least NoVars variables that meet the previous criterion, then we shift Anno i by $BestLag_i$. If we have less than NoVars variables, then we go back to (2) and find the next best lag and variable.
The corresponding algorithm is:

Alg3: Best Lag, Chosen Variables
Input: k variables, n accepted annotations/motifs, Main annotation, IgnoredVars and set parameters: MaxLag Output: Best Lag, Chosen Variables for each motif except MainAnno CheckedLags={} For each motif i: While $len(ChosenVars_i) < NoVars$ and len(CheckedLags)<MaxLag: $Lag_i, Var_i = \text{argmax}_{lag=-ML,ML,var=1,k} Corr(Anno\ i, MainAnno)$ (2) CheckedLags.append(Lag_i) For each other variables v not in IgnoredVars: If $Corr(Anno\ i_v, MainAnno_v) > MinCorr$: $ChosenVars_i.append\ (k1)$ If $len(ChosenVars_i) >= NoVars$: Return ChosenVars else: Choose a different lag
Return ChosenVars

If there exists an annotation i for which there are less than NoVars significant variables for any lag, that annotation is considered as an outlier[2].

[2] an outlier in this case is an erroneous annotation by the user which we isolated by not correlating well with the other annotations we have for the same event

After shifting all annotations (except the main one which remains unchanged), the list of variables describing best all the annotations is determined by intersecting the individual lists of all motifs except outlier motifs.

2.3 Example test data and processing steps

The previous procedure was tested on sensor data consisting of 94 variables covering approximately one week of data, sampled at one second. In the experiment, an user has marked a frequent event by giving 4 annotations (motifs) of occurrences. These annotations differed in length (from 7 minutes to 10 minutes) and in their alignment. The huge, visible variations between the example annotations can be caused by different reasons: an user might or might not include a certain time band before the event or after, so it is never clear what the the pattern itself actually is, e.g. where exactly it starts or ends.
Visually, the 4 time areas selected by the user look as follows (for one of the variables):

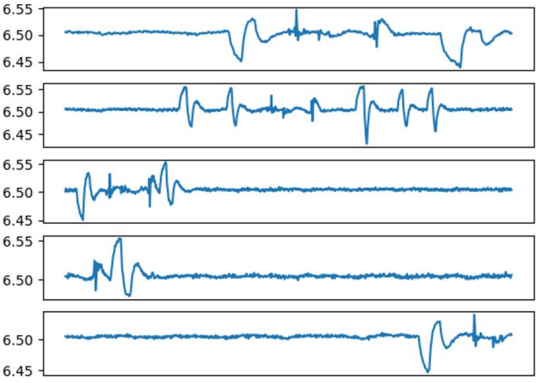

Figure 1: Initial user annotations

Although the series seem to have a similar behavior and the range of values they cover are similar, it's difficult to pinpoint exactly how they should be aligned.
After running the analysis, the main motif and the modified motifs show more synchronicity, as shown below:

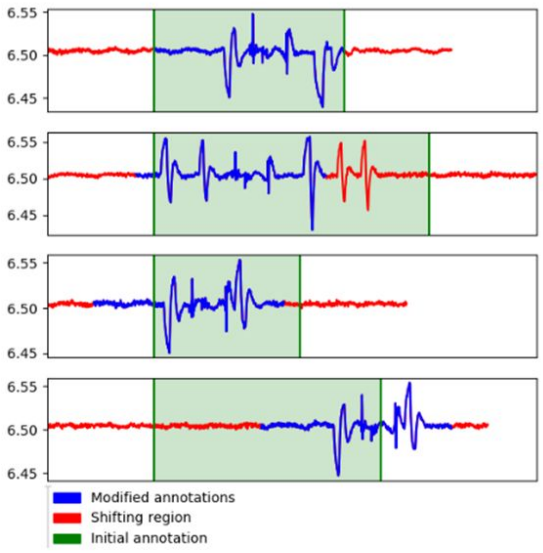

Figure 2: Shifting procedure

The first motif is the only one that remains unchanged and represents the main motif, while the others are shifted to the left/right as indicated by the blue lines. The red lines indicate the maximum shift possible.

The result of the shift can be better assessed by comparing the cross-correlation between the motifs before and after shifting. The initial correlation results are calculated as an average correlation for the 5 variables that are selected as best by the algorithm. The results are summarized below:

	$Motif_1$	$Motif_2$	$Motif_3$	$Motif_4$
$Motif_1$	1	0,4082	-0.1298	-0.4039
$Motif_2$	0,4082	1	0.0002	-0.3733
$Motif_3$	-0.1298	0.0002	1	-0.4293
$Motif_4$	-0.4039	-0.3733	-0.4293	1

The correlation between the modified motifs is summarized in the table below:

	MainMotif	$ModMotif_2$	$ModMotif_3$	$ModMotif_4$
MainMotif	1	0.9845	0.9969	0.9056
$ModMotif_2$	0.9845	1	0.4104	0.3656
$ModMotif_3$	0.9969	0.4104	1	-0.1627
$ModMotif_4$	0.9056	0.3656	-0.1627	1

The average cross-correlation has increased for all combinations of motifs, but the biggest increase can be seen between the main motif and all other motifs,which is close to 1.. The smallest increase can be noted between Motif3 and Motif4, where the cross-correlation, despite the increase, remains negative.

3 Results

The method described in the previous section proves to be an efficient tool in the preprocessing step necessary to any pattern miner. More specifically, having only an indication of start times and end times of an event and a series of variables (not necessarily relevant in describing the event) , the algorithm aligns the annotations in such a way that they show the same event and indicates which variables to take into consideration when building the pattern miner.

For example, SAX is an univariate pattern miner that can be extended to the multivariate case by aggregating the individual distances obtained for each univariate miners on a set of variables. Without a prior selection of the variables, using an aggregating measure as the average can introduce noise into the process and lead to erroneous results. Moreover, SAX mines for one pattern at a time, so providing the best event example is vital.

Using the dataset described in section 2.3, and mining for Motif 1 (which was identified by the algorithm as the main motif) the comparative results of the algorithm look in the following way:

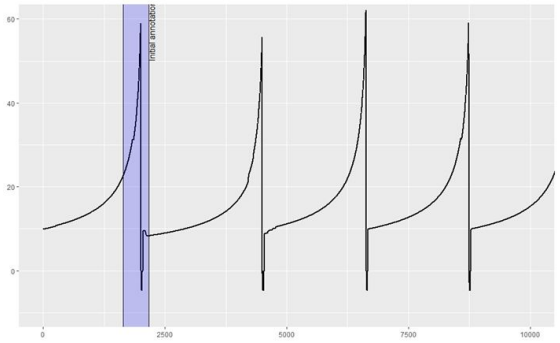

Figure 3: SAX with all the variables

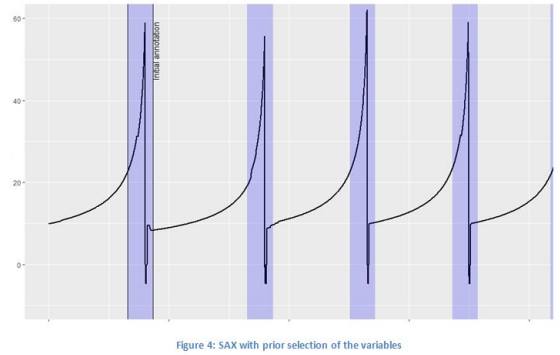

Figure 4: SAX with prior selection of the variables

The miner that what given the entire dataset as input has difficulties in correctly identifying the pattern (marked with vertical lines), leading to no identified regions. On the other hand, the second miner could identify all occurences of the patten. Moreover, the running time is significantly

improved with the elimination of irrelevant variables.
Another important aspect that the proposed algorithm looks into is the possibility to handle negative annotations. This is done by identifying and ignoring variables that behave similarly for negative annotations and correct annotations. For example, if in the previous setting another annotation that represents a negative example would have been introduced, all the variables for which the main motif is highly correlated with the negative motif would be excluded from the list of important variables.

4 Conclusion

In this paper, an algorithm for supporting user annotations for multivariate motif mining was proposed. The method consisted of adjusting the annotations provided by experts as to achieve the best similarity, while identifying incorrect annotations and the variables that best describe the correct ones. Moreover, in the variable selection process, negative annotations given by the user are taken into consideration, by indicating the variables that are misleading and should be ignored.
The algorithm should be seen as a preprocessing step for pattern miners, as it was shown that it can help them identify true patterns by eliminating noise introduced by some variables.

5 Outlook

In two cases we are using the 75-percentile as an aggregation function to reduce the dependencies on parameters (variables, lags). There are other ways of aggregations possible like mean, median, max. We selected this aggregation to achieve robust results based on a pragmatic intuition and experimental results:
As we base all preprocessing and selection steps on correlations, taking e.g. the max as aggregation could easily mislead the results towards an exceptional good correlation between the motifs on a certain variable or a certain time whereas all other correlations might be very bad. The opposite behaviour is true for the mean: if there is a good portion of bad correlations, they will shift the mean towards a low value, which will e.g. later be used as an acceptance threshold and will then include too many "bad areas". The selection of a "good" aggregation function is still an important part in the procedure; there is still room to automate this step based on a quality measure or other automatic approach to find a good aggregations as a good selection might differ between applications.

References

Berndt, Donald & Clifford, James. 1994. "Using Dynamic Time Warping to Find Patterns in Time Series", KDD workshop. 10/16. 359-370.

Gao, Y. & Lin, J., 2018. Exploring variable-length time series motifs in one hundred million length scale. *Data Mining and Knowledge Discovery*, 32(5), pp.1200–1228. Available at: http://dx.doi.org/10.1007/s10618-018-0570-1.

Keogh, E. et al., An online algorithm for segmenting time series. *Proceedings 2001 IEEE International Conference on Data Mining*. Available at: http://dx.doi.org/10.1109/icdm.2001.989531.

Lin, J. et al., 2007. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2), pp.107–144. Available at: http://dx.doi.org/10.1007/s10618-007-0064-z.

Appendix

$$Corr(Anno\ i, Anno\ j) = \begin{bmatrix} Corr(Y_{1,[t_{i1},t_{i2}]}, Y_{1,[t_{j1-ML},t_{j1-ML+LenAnno\ i}]}) & Corr(Y_{2,[t_{i1},t_{i2}]}, Y_{2,[t_{j1-ML},t_{j1-ML+LenAnno\ i}]}) & \cdots & \cdots & \cdots & Corr(Y_{k,[t_{i1},t_{i2}]}, Y_{k,[t_{j1-ML},t_{j1-ML+LenAnno\ i}]}) \\ Corr(Y_{1,[t_{i1},t_{i2}]}, Y_{1,[t_{j1-ML+1},t_{j1-ML+1+LenAnno\ i}]}) & Corr(Y_{2,[t_{i1},t_{i2}]}, Y_{2,[t_{j1-ML+1},t_{j1-ML+1+LenAnno\ i}]}) & \cdots & \cdots & \cdots & Corr(Y_{k,[t_{i1},t_{i2}]}, Y_{k,[t_{j1-ML+1},t_{j1-ML+1+LenAnno\ i}]}) \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ Corr(Y_{1,[t_{i1},t_{i2}]}, Y_{1,[t_{j1+ML-1},t_{j1+ML-1+LenAnno\ i}]}) & Corr(Y_{2,[t_{i1},t_{i2}]}, Y_{2,[t_{j1+ML-1},t_{j1+ML-1+LenAnno\ i}]}) & \cdots & \cdots & \cdots & Corr(Y_{k,[t_{i1},t_{i2}]}, Y_{k,[t_{j1+ML-1},t_{j1+ML-1+LenAnno\ i}]}) \\ Corr(Y_{1,[t_{i1},t_{i2}]}, Y_{1,[t_{j1+ML},t_{j1+ML+LenAnno\ i}]}) & Corr(Y_{2,[t_{i1},t_{i2}]}, Y_{2,[t_{j1+ML},t_{j1+ML+LenAnno\ i}]}) & \cdots & \cdots & \cdots & Corr(Y_{k,[t_{i1},t_{i2}]}, Y_{k,[t_{j1+ML},t_{j1+ML+LenAnno\ i}]}) \end{bmatrix}$$

In the matrix above, each entry represents the correlation of two sections in time of one variable. The two sections in time are derived from the user annotation Anno i and Anno j. The first section is the first Annotation (Anno i) in its original unshifted version (as annotated by the user) and the second section is the Annotation j which is adjusted in length according to the first Annotation and then shifted over the time area [-ML,+ML]. Each row represents the values of the correlations at one specific lag, across all variables. Each column represents the values of the correlation for a specific variable, across all lags within the [-ML,+ML] time adjustments. Since there is one alignment that is superior to all others, we expect to see one value on each column that is significantly higher than the others (at least for the variables that are representative in describing the pair of motifs). If the user made perfect aligned annotations, this will occur at time shift 0.
Maximizing across lags will give one value per each variable, as shown below:

$$\max_{lag=-ML,ML} Corr(Anno\ i, Anno\ j) = \\ = \left[\max_{lag=-ML,ML} Corr\left(Y_{1,[t_{i1},t_{i2}]}, Y_{1,[t_{j1+lag},t_{j1+lag+LenAnno\ i}]}\right) \quad \cdots \quad \cdots \quad \cdots \quad \max_{lag=-ML,ML} Corr\left(Y_{k,[t_{i1},t_{i2}]}, Y_{k,[t_{j1+lag},t_{j1+lag+LenAnno\ i}]}\right)\right]$$

Because we expect a state to be described by more than one variable, we avoid taking the maximum as an aggregation method for the values obtained at the previous step (this makes sure that we don't choose a motif solely on one variable). Also, some of the k variables might be noisy, irrelevant in describing the motifs so a method that is robust to outliers is necessary. Aggregating across variables through the 75th percentile will give one value per pair of motifs:

$$Aggregated\ Corr(Anno\ i, Anno\ j) = 75th\ Percentile\ of \max_{lag=-ML,ML} Corr(Anno\ i, Anno\ j)$$

This gives j-1 values for each Annotation, and the overall correlation for each annotation is obtained as a percentile of all previous values.