Efficient Computation of All-Window Length Correlations

Adam Charane, Matteo Ceccarello, Anton Dignös, Johann Gamper

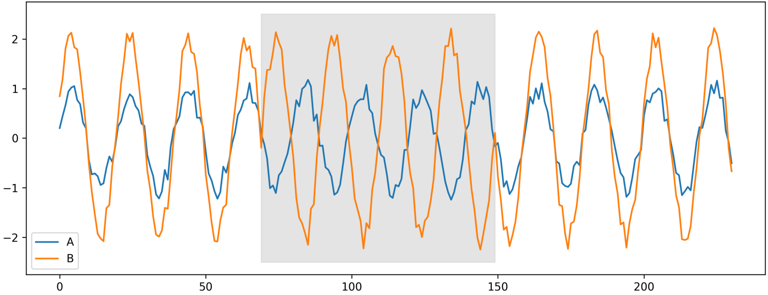
Free University of Bozen-Bolzano

Wednesday 6th July, 2022

Supported by the European Regional Development Fund - Investment for Growth and Jobs Programme 2014–2020. Project PREMISE (FESR1164).

Motivation and Contributions

Motivation



1. At which time the behavior between the two time series changes ?
2. For how long does this new behavior lasts ?

Contributions

Our contributions can be summarized as:

1. Providing correlations over all possible subsequences for analysts to analyze.
2. Providing a visual summarization of all the correlations.
3. Efficiently computing the correlations by caching overlapping computations.
4. Speeding up computations by exploiting cache memory and parallelization.

Definitions & Use Case Examples

Definitions

Definition (Time Series)

A Time series T is an ordered sequence of measurements from a process: $T = t_1, t_2, \ldots, t_n$

Definition (Time Series Subsequences)

A subsequence $T_{i,w}$ is a consecutive subset from T starting at i and having w measurements: $T_{i,w} = t_i, t_{i+1}, \ldots, t_{i+w-1}$

Definition (Pearson correlation)

The Pearson correlation coefficient of two vectors T and S is defined as: $\rho_{T,S} = \frac{\mathbb{E}[(T-\mu_T)(S-\mu_S)]}{\sigma_T \sigma_S}$

Problem Definition

Definition (All-Window Length Correlations Set)

Given two time series T and S of equal length n, the all-window length correlations set is defined as $\{\rho_{T_{i,w},S_{i,w}} \big| i \in [1 \,..\, n-1] \wedge w \in [2 \,..\, n] \wedge i + w \leq n\}$

Example

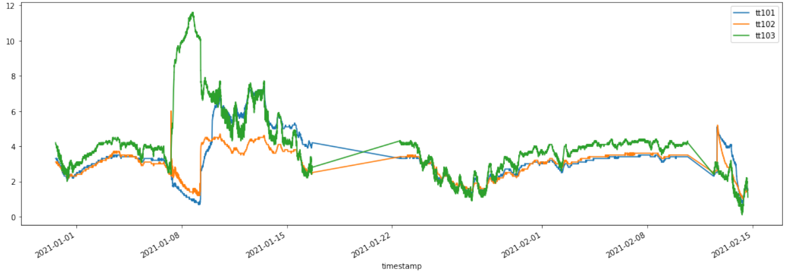


Figure: Three time series over a period of a month and a half

In the above example, we have:

- The correlation between tt101 and tt103 is 0.2.
- If we ignore the of 8 January, the correlation becomes 0.93.

Heatmap Example

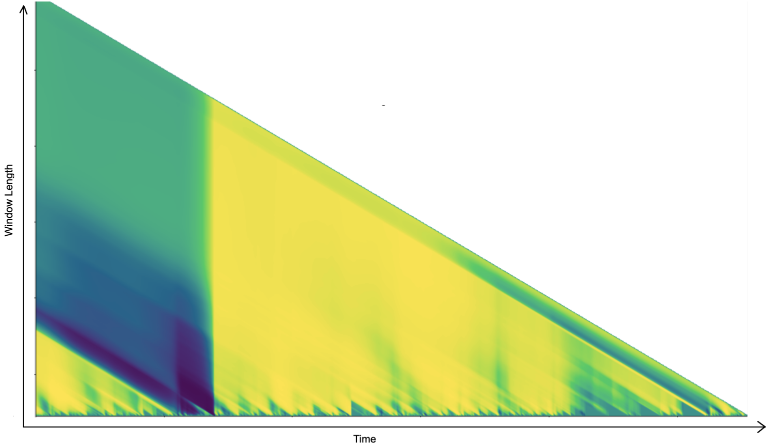


Figure: Heatmap of correlation between tt101 and tt103.

Heatmap Example

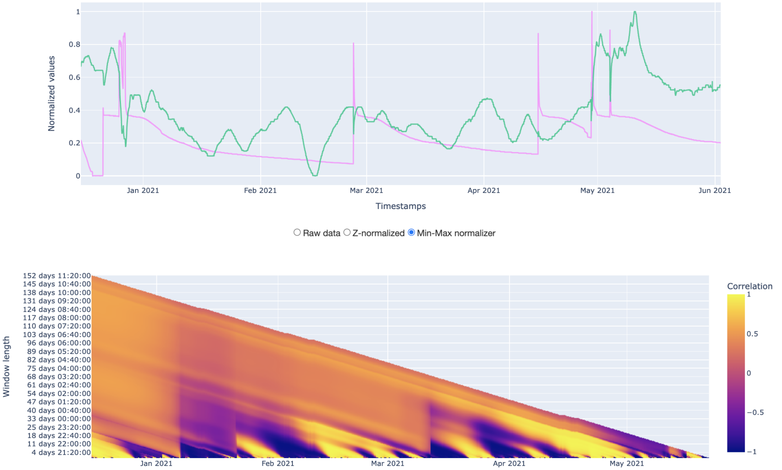


Figure: Heatmap in interactive mode.

Computation of All-Window Length Correlation Set

Incremental Computation

By rewriting Pearson coefficient between subsequences as follows [1]:

$$\rho_{T_{i,w},S_{i,w}} = \frac{\mathbb{E}[(T_{i,w} - \mu_{T_{i,w}})(S_{i,w} - \mu_{S_{i,w}})]}{\sigma_{T_{i,w}}\sigma_{S_{i,w}}} \quad (1)$$

$$= \frac{n\sum_{j=i}^{i+w-1} T_j S_j - \sum_{j=i}^{i+w-1} T_j \sum_{j=i}^{i+w-1} S_j}{\sqrt{n\sum_{j=i}^{i+w-1} T_j^2 - (\sum_{j=i}^{i+w-1} T_j)^2}\sqrt{n\sum_{j=i}^{i+w-1} S_j^2 - (\sum_{j=1}^{i+w-1} S_j)^2}} \quad (2)$$

We can do the computation in an incremental fashion.

Incremental Computation

Let's define the quantities:

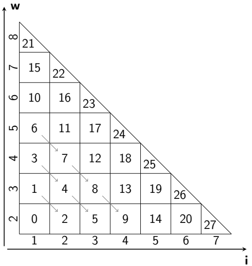
- $T_{\Sigma}^{(i,w)} = \sum_{j=i}^{i+w-1} T_j$
- $S_{\Sigma}^{(i,w)} = \sum_{j=i}^{i+w-1} S_j$
- $T_{\Sigma^2}^{(i,w)} = \sum_{j=i}^{i+w-1} T_j^2$
- $S_{\Sigma^2}^{(i,w)} = \sum_{j=i}^{i+w-1} S_j^2$
- $TS_{\Sigma}^{(w,)} = \sum_{j=i}^{i+w-1} T_j S_j$

Then Pearson correlation can be expressed as:

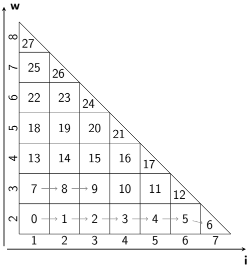
$$\rho_{T_{i,w},S_{i,w}} = \frac{n \cdot TS_{\Sigma}^{(i,w)} - T_{\Sigma}^{(i,w)} \cdot S_{\Sigma}^{(i,w)}}{\sqrt{nT_{\Sigma^2}^{(i,w)}} \cdot \sqrt{nS_{\Sigma^2}^{(i,w)}}} \tag{3}$$

Memory Layout

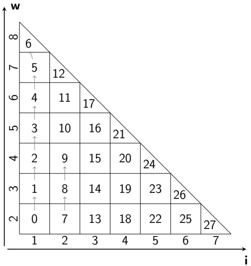
In order to save space and exploit cache memory, we store the correlations in an array, and we propose three different memory layouts:



(a) Anti-Diagonal



(b) Horizontal



(c) Vertical

Figure: Linearization of the matrix for the different memory layouts for two time series of dimension 8.

Speed-ups

Based on the following three remarks:

- ⇒ Each two consecutive anti-diagonals are also consecutive in the array.
- ⇒ Each two consecutive values in a diagonal are represented as consecutive values in the array.
- ⇒ Each value depends on its previous one (update rule).

Speed-ups

We can conclude:

- ⇒ Computation using the update rule is cache friendly.
- ⇒ Each number of consecutive diagonals can be computed in parallel.

Experimental Evaluation

Scalability

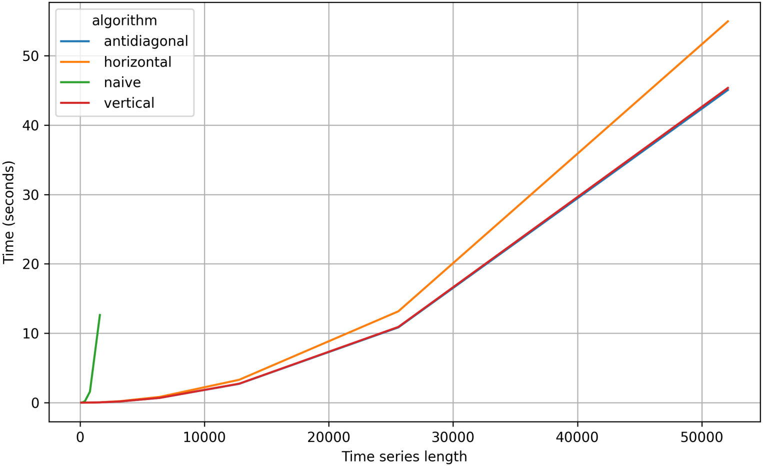


Figure: Scalability of Naive approach vs our method.

Parallelism

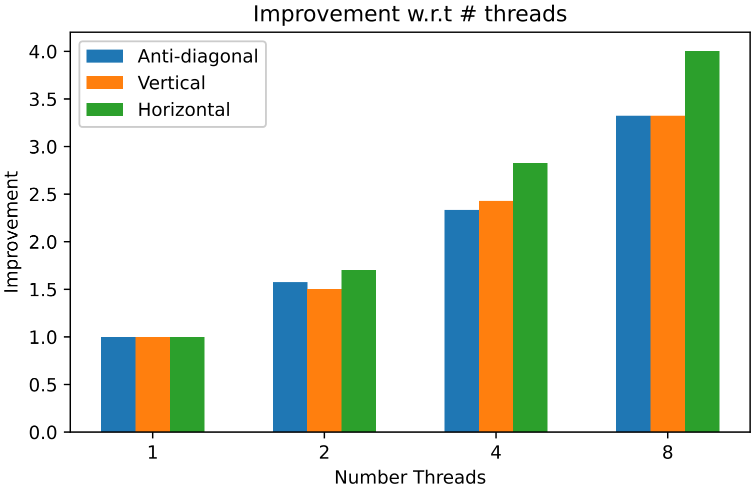


Figure: The effect of number of threads on the computation.

Conclusions

Conclusions

$\Rightarrow$ We find all phenomenons between two time series.

$\Rightarrow$ We provide an efficient method to compute correlations.

$\Rightarrow$ We provide a visual summary to easily identify interesting phenomenons.

Thank you !

References

BRAID: Stream Mining through Group Lag Correlations

Incremental computation - Update Rule

Assume we have $T_\Sigma^{(i,w)}$. Then, we can compute in constant time both $T_\Sigma^{(i+1,w)}$ and $T_\Sigma^{(i,w+1)}$:

$$T_\Sigma^{(i,w+1)} = T_\Sigma^{(i,w)} + T_{i+w} \tag{4}$$

$$T_\Sigma^{(i+1,w)} = T_\Sigma^{(i,w)} + T_{i+w} - T_i \tag{5}$$

By applying the same update to the other quantities, we can compute the all-window correlation set in $\mathcal{O}(n^2)$