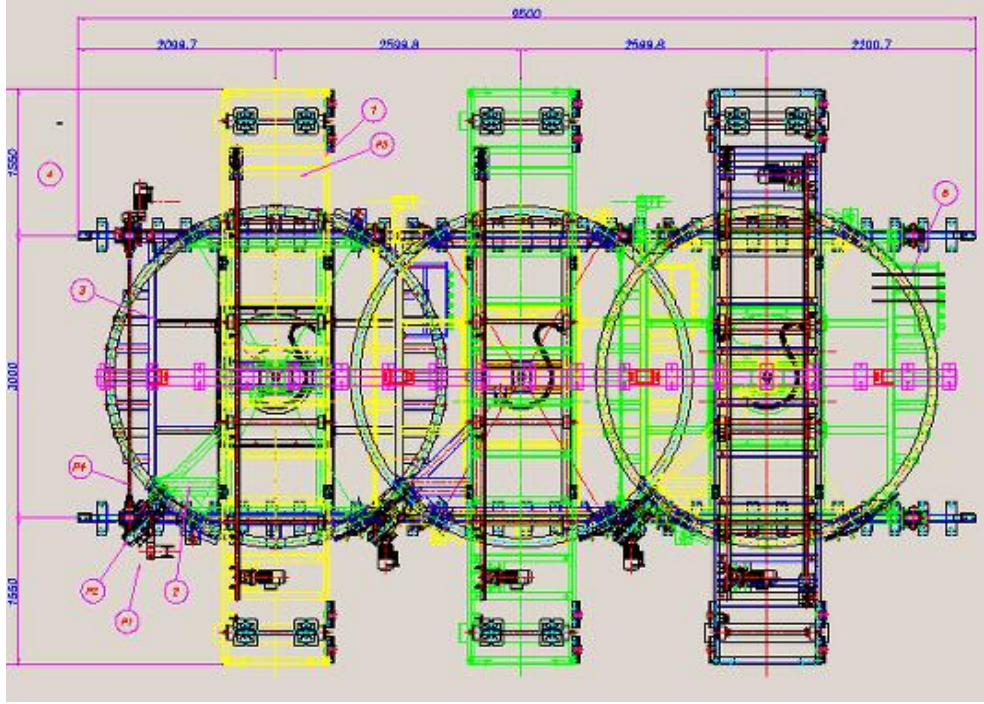
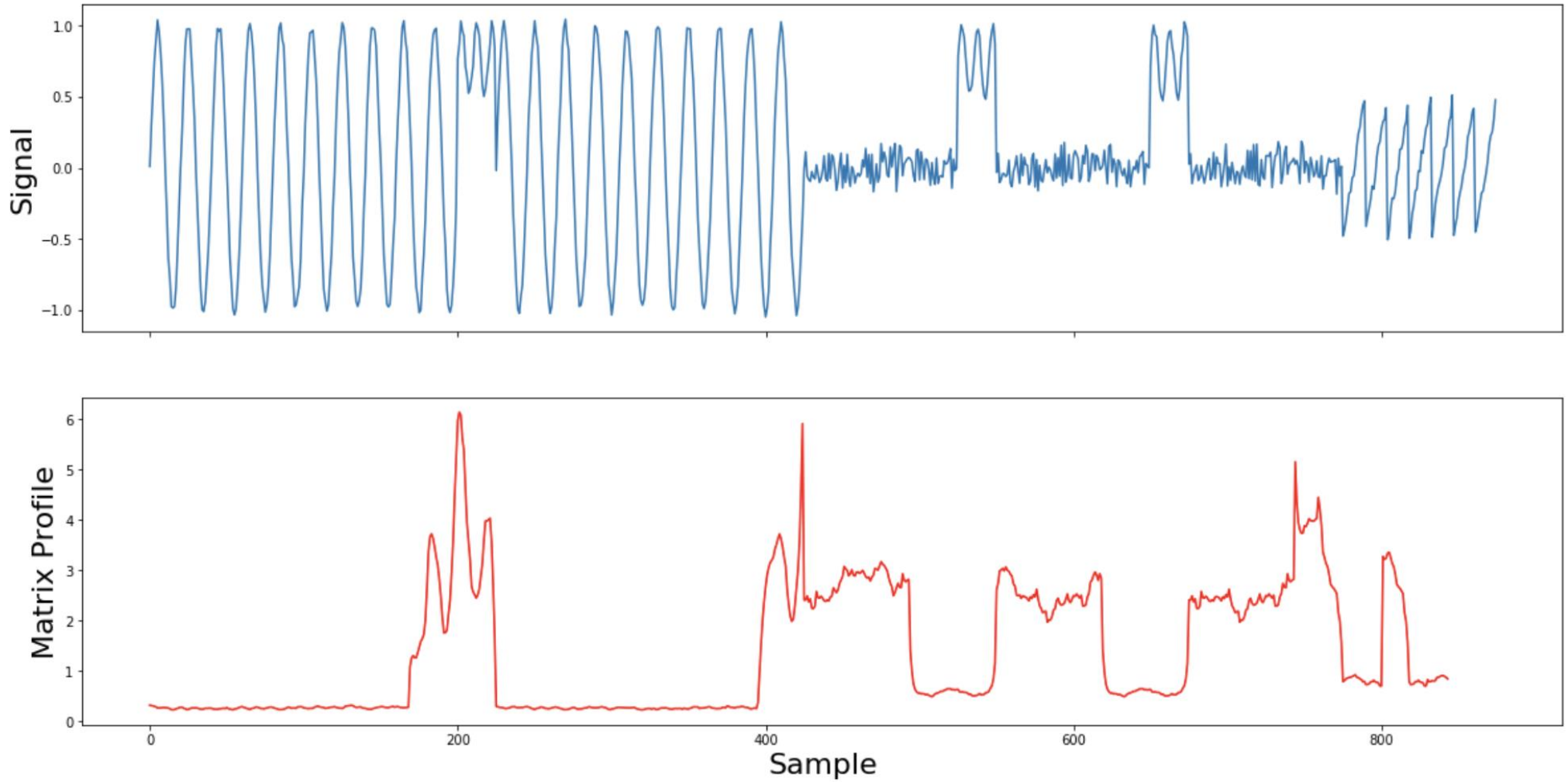

9500
2599.7
2599.8
2599.8
2700.7
1550
3000
1550

1-d Real Value data 를 입력하고,
내부적으로 Distance Matrix 를 계산하고,
이 값들의 Min 을 가져오면 Matrix Profile 이 된다.
(Min 이외에 Matirx Profile Index, Motif, Discords 를 찾을 수 도 있다.
https://gitlab.com/wooheaven/Python-Study/blob/master/06_MatrixProfile/02_matrixprofile-ts/01_Matrix_Profile_Tutorial.ipynb



This tutorial is funded by:

- NSF IIS-1161997 II
- NSF IIS 1510741
- NSF 544969
- CNS 1544969
- SHF-1527127
- AFRL FA9453-17-C-0024

https://www.cs.ucr.edu/~eamonn/MatrixProfile.html

Any errors or controversial statements are due solely to Mueen and Keogh

Oracle IoT : https://www.datascience.com/blog/sax-and-matrix-profile-time-series

Making SAX Efficient With Matrix Profile

While SAX met our goals to identify anomalous events of interest and search for them, we realized that we had to tune parameters for word length and alphabet size. In addition, to ensure that we could track these events in the future, what we wanted was the ability to search for these patterns in real-time. For this, we built Matrix Profile, which computes a companion time series data set. Matrix Profile, for a sequence, records the distance to its most similar neighbor. For example, the sequence starting at 921 has its nearest neighbor at a distance of 177.

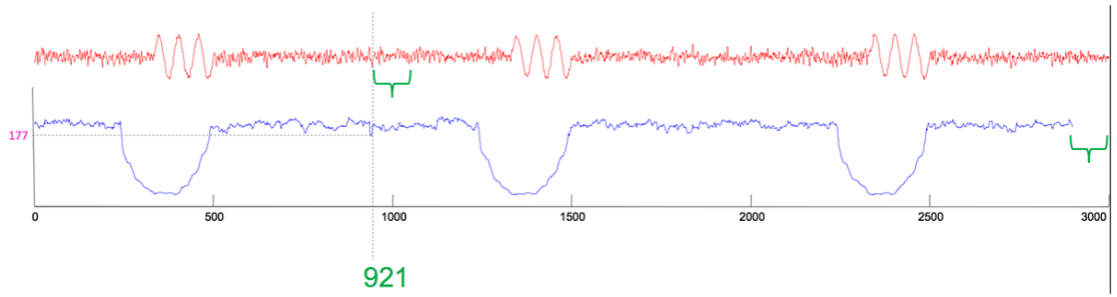


Figure 4: Matrix Profile is a companion time series that specifies distance to the similar sequence

Behind Story on UCR Time Series on Similarity Search : DTW -> Matrix Profile

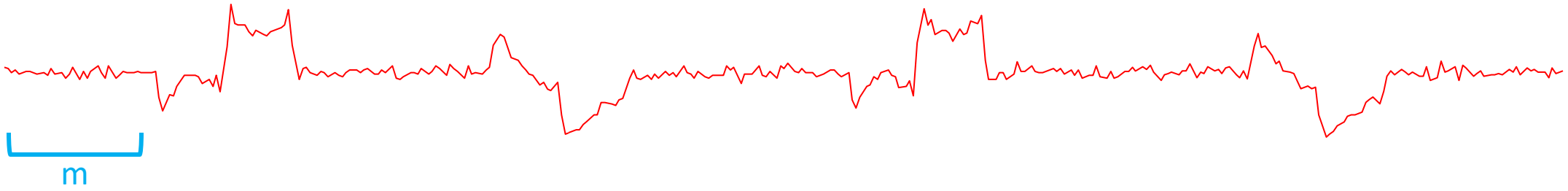
http://practicalquant.blogspot.com/2012/10/mining-time-series-with-trillions-of.html
https://franzbischoff.github.io/tsmp/

Motivation

Note that for most time series data mining tasks, we are not interested in any *global* properties of the time series, we are only interested in small *local* subsequences, of this length, *m*

These subsequences might be about the length of individual heartbeats (for ECGs), individual days (for social media behavior), individual words (for speech analysis) etc

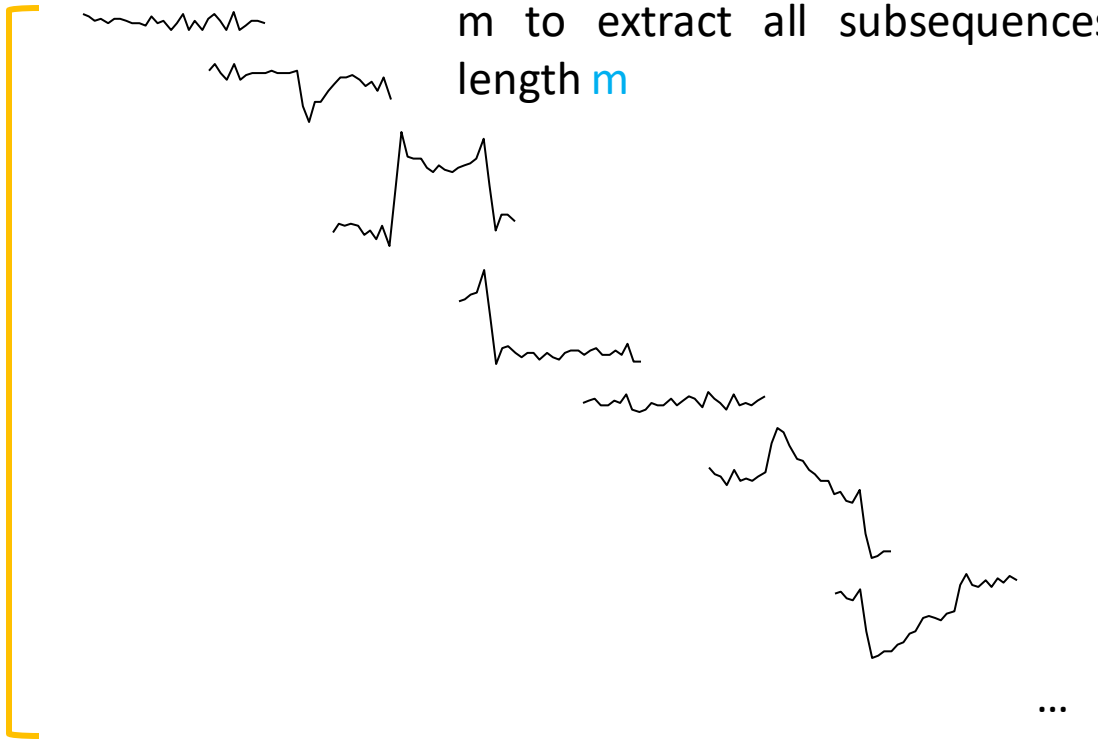
Given a time series, T and a desired subsequence length, m



Motivation

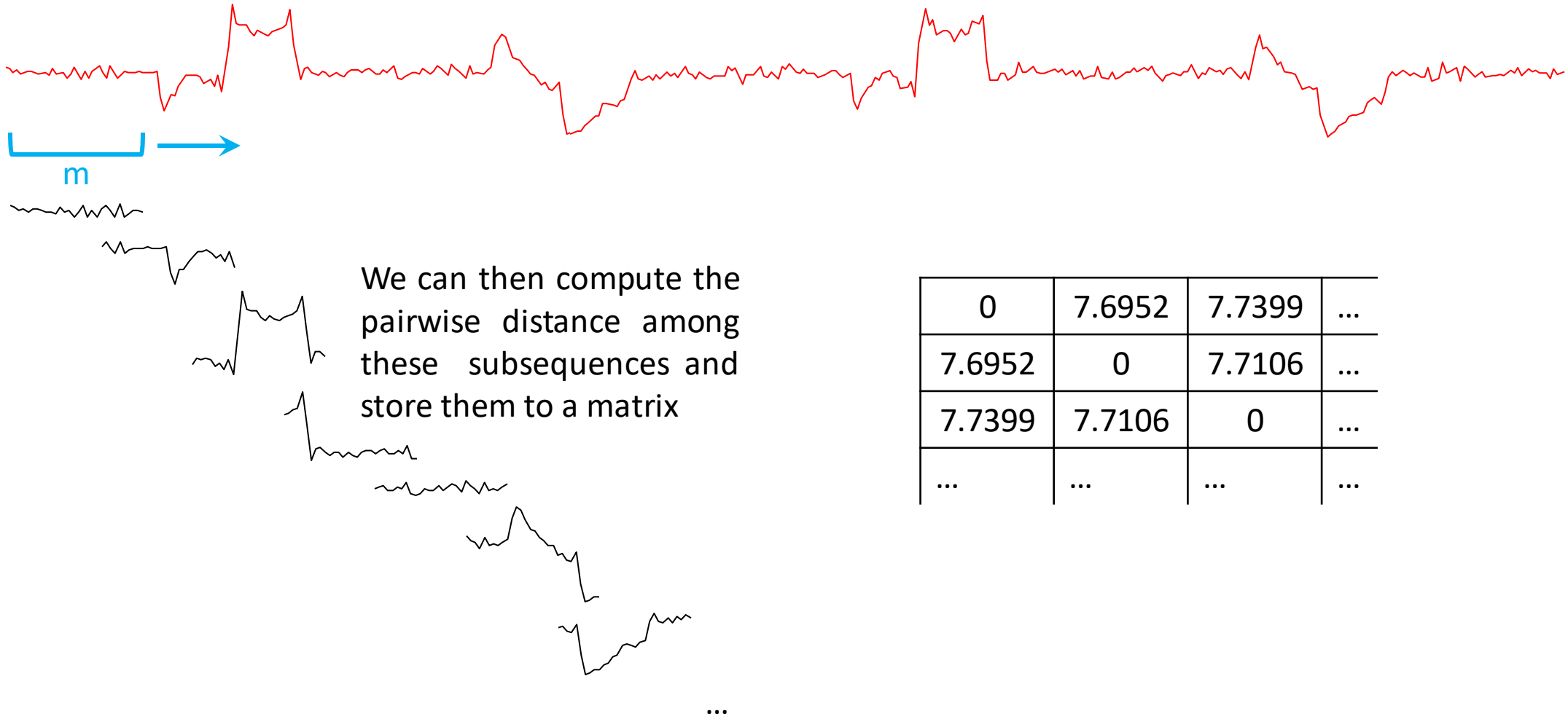
Given a time series, T and a desired subsequence length, m

We can use sliding window of length m to extract all subsequences of length m



Motivation

Given a time series, T and a desired subsequence length, m



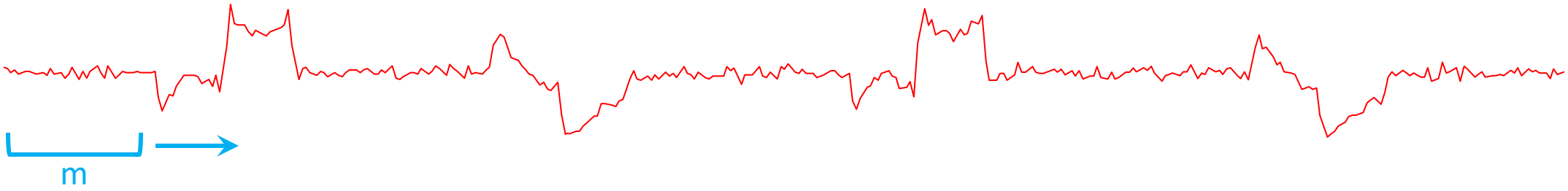
We can then compute the pairwise distance among these subsequences and store them to a matrix

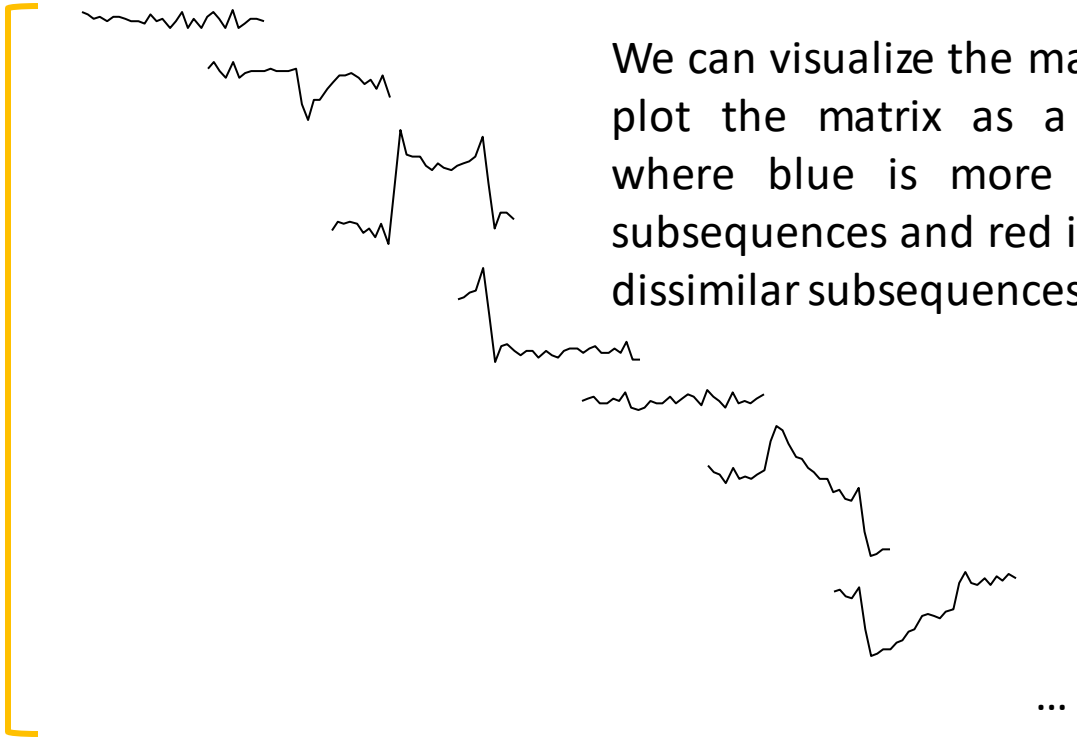
0	7.6952	7.7399	…
7.6952	0	7.7106	…
7.7399	7.7106	0	…
…	…	…	…

9

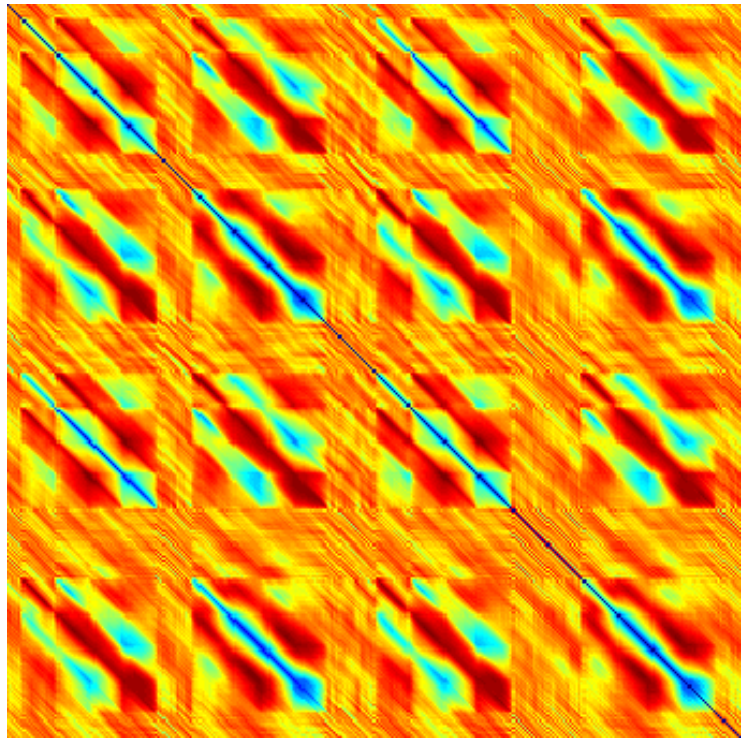
Motivation

Given a time series, T and a desired subsequence length, m





We can visualize the matrix by plot the matrix as a image where blue is more similar subsequences and red is more dissimilar subsequences



10

This equation has 3-pain-point.

$$Prove\ this\ equation\ :\ d_{i,\ j} = \sqrt{2m\left(1 - \frac{QT_{i,\ j} - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)}$$

$$By\ definition\ :\ (d_{i,\ j})^2 = \sum_{k=0}^{m-1}\left\{\left(\frac{T_{i,\ m} - \mu_i}{\sigma_i}\right) - \left(\frac{T_{j,\ m} - \mu_j}{\sigma_j}\right)\right\}^2$$

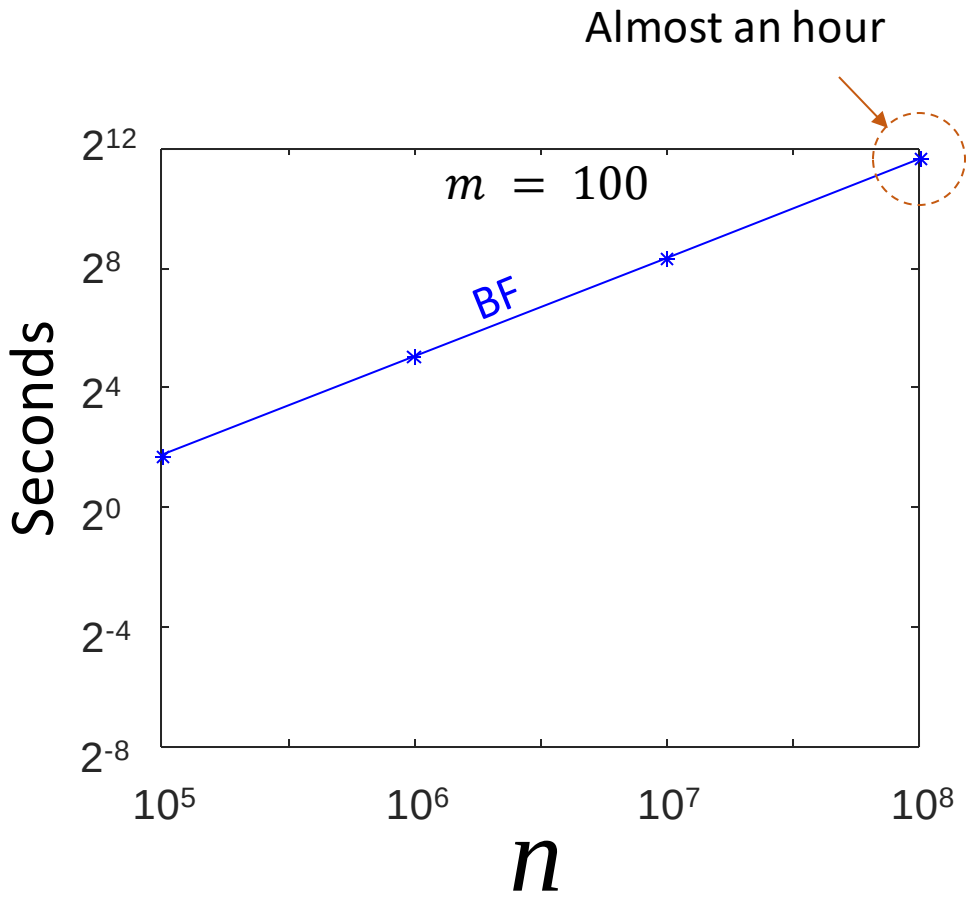
https://github.com/wooheaven/PlayOnSandBox/blob/master/02_MP/01_matrixprofile-rwoo/MP_derived_distance_equation.ipynb

The Brute Force Algorithm

- Scan the time series with a sliding window
- Z-Normalize the window
- Calculate Euclidean distance between window and the query

Why

```
d(1:n) = 0;
Q = zNorm(query);
for i = 1:n-m+1
    d(i) = sqrt(sum((zNorm(T(i:i+m-1))-Q).^2));
end
```



- How long does this algorithm take?
- The time complexity is $O(nm)$ in the average and worst cases. More precisely the window is scanned two times in each iteration of this algorithm. One scan is for z-normalization, and the other scan is for distance calculation.
- Note that we cannot use any early abandoning or pruning as we need all the distances.

Z-Normalize is ok : ECG hart pattern

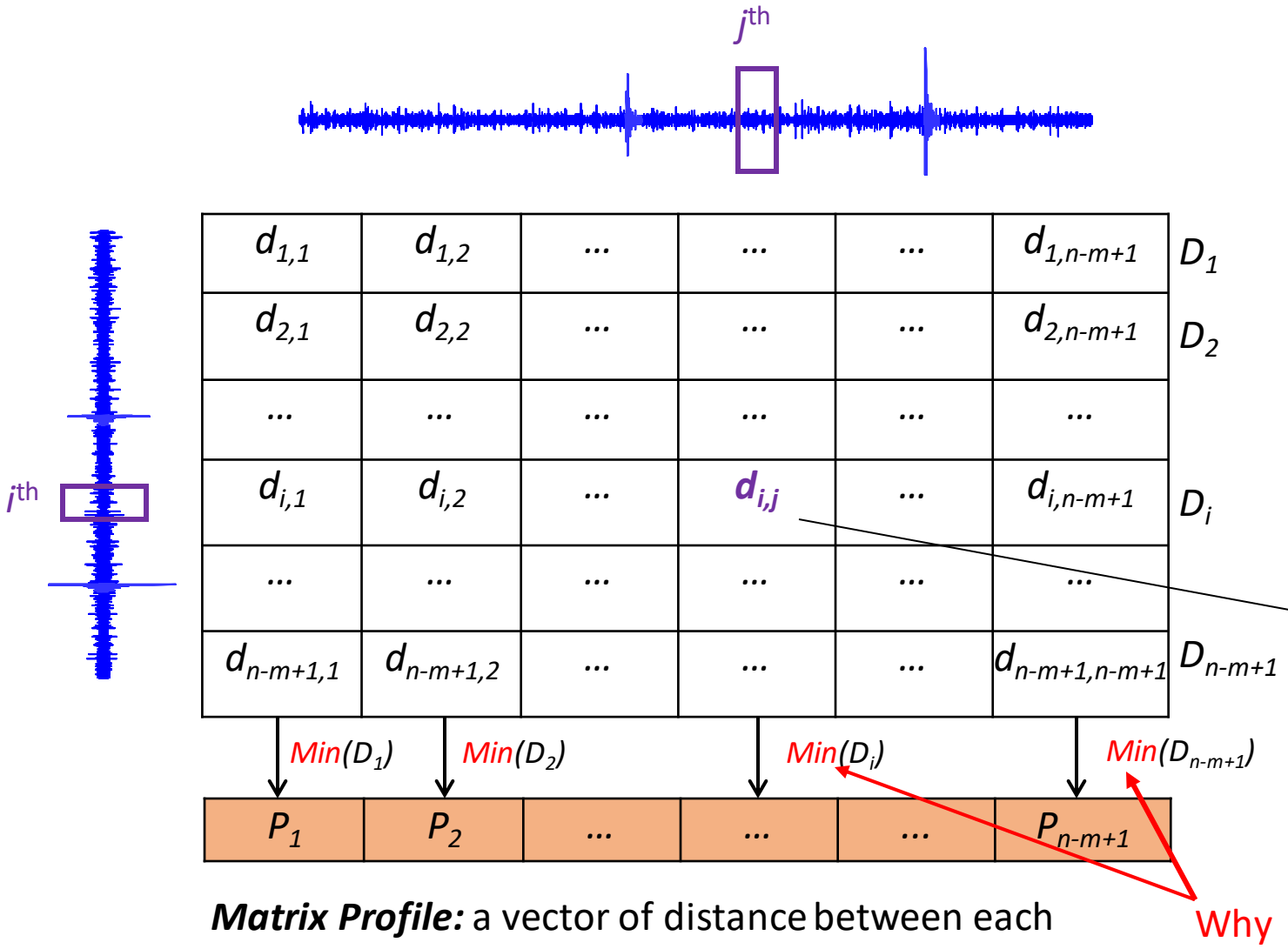
https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitor-hookup-guide

http://ceur-ws.org/Vol-2322/DARLIAP_10.pdf

Z-Normalize is not ok : Stock Pattern

https://m.blog.naver.com/dngineer/221506565764

Matrix Profile from Distance Profiles



The distance matrix is symmetric
The diagonal is zero
Cells close to the diagonal are very small

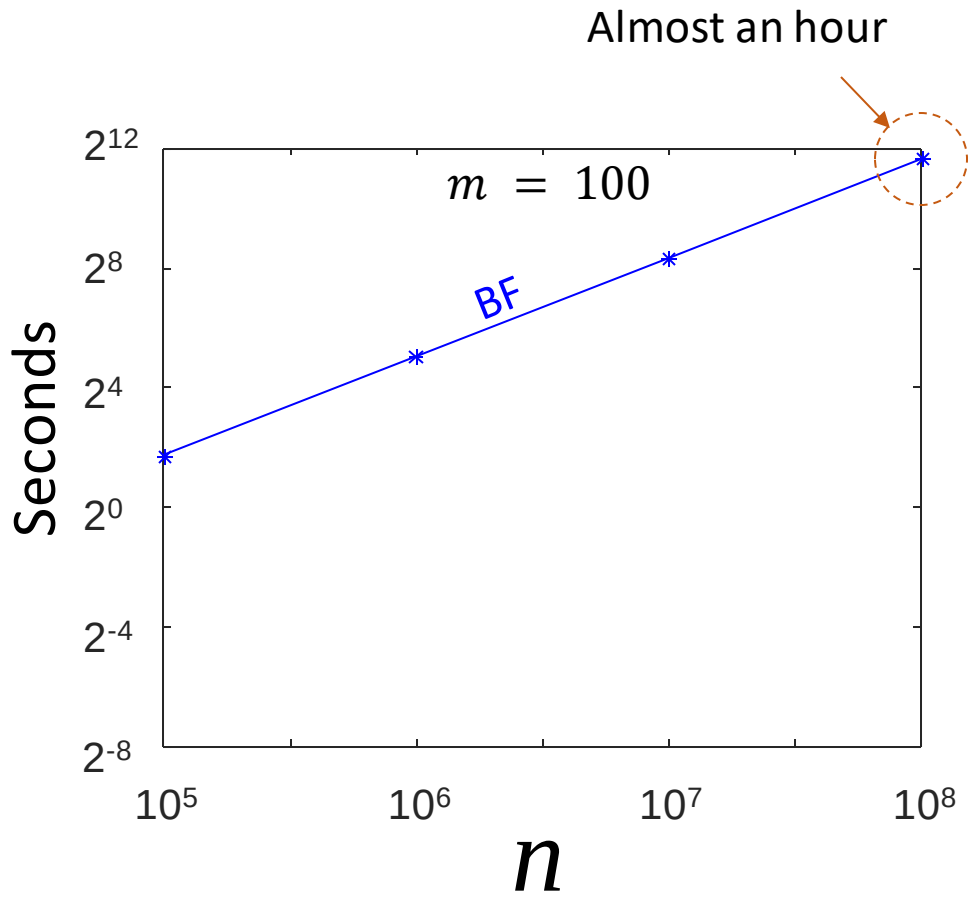
$d_{i,j}$ is the distance between the i^{th} window and the j^{th} window of the time series

Matrix Profile: a vector of distance between each subsequence and its nearest neighbor

The Brute Force Algorithm

- Scan the time series with a sliding window
- Z-Normalize the window
- Calculate Euclidean distance between window and the query

```
d(1:n) = 0;
Q = zNorm(query);
for i = 1:n-m+1
    d(i) = sqrt(sum((zNorm(T(i:i+m-1))-Q).^2));
end
```



- How long does this algorithm take?
- The time complexity is $O(nm)$ in the average and worst cases. More precisely the window is scanned two times in each iteration of this algorithm. One scan is for z-normalization, and the other scan is for distance calculation.
- Note that we cannot use any early abandoning or pruning as we need all the distances.

Just-in-time Normalization (1 of 3)

- Can we skip the z-normalization scan in each iteration?
- Yes, if we have the means, standard deviations and the dot product to calculate the distances.
- z-normalized sequence has zero mean and one standard deviation.
- Let's assume $\boldsymbol{y}$ is the z-normalized query, and $\boldsymbol{x}$ is the time series (T), therefore, $\mu_y = 0$ and $\sigma_y = 1$

Working Formula

$$d(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \sqrt{2m\left(1 - \frac{\sum_{i=1}^{m} x_i y_i - m\mu_x\mu_y}{m\sigma_x\sigma_y}\right)}$$

$$d(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \sqrt{2\left(m - \frac{\sum_{i=1}^{m} x_i y_i}{\sigma_x}\right)}$$

Thanawin Rakthanmanon, et al. Searching and mining trillions of time series subsequences under dynamic time warping. KDD 2012: 262-270

Just-in-time Normalization (2 of 3)

- Can we skip the z-normalization scan in each iteration?
- The standard deviations of moving windows of a fixed size can be calculated in one linear scan.
  - In one pass, calculate cumulative sums of x and x^2 and store
  - Subtract two cumulative sums to obtain the sum over any window
  - Use the sums to calculate the standard deviations of all windows in linear time
- In 2016, MATLAB has introduced a function, `movstd`, that does the above.

$$d(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \sqrt{2\left(m - \frac{\sum_{i=1}^{m} x_i y_i}{\sigma_x}\right)}$$

$$C = \sum x \qquad C^2 = \sum x^2$$

$$S_i = C_{i+m} - C_i \qquad S_i^2 = C_{i+m}^2 - C_i^2$$

$$\sigma_i = \sqrt{\frac{S_i^2}{m} - \left(\frac{S_i}{m}\right)^2}$$

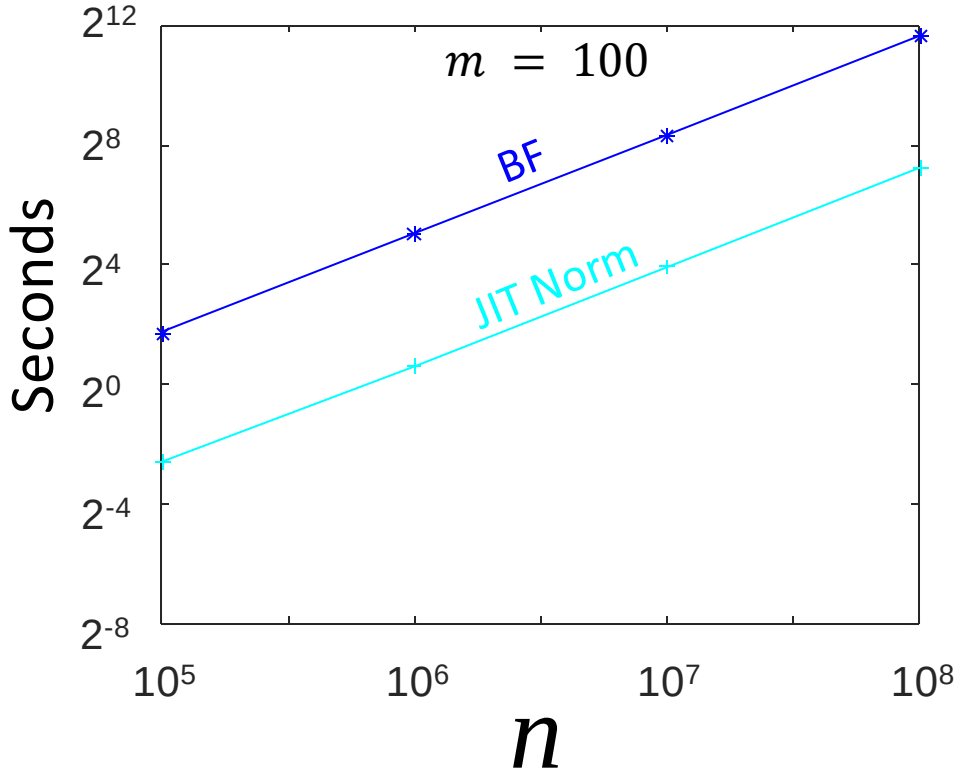
Thanawin Rakthanmanon, et al. Searching and mining trillions of time series subsequences under dynamic time warping. KDD 2012: 262-270

Just-in-time Normalization (3 of 3)

- Can we skip the z-normalization scan in each iteration?

```
d(1:n) = 0;
Q = zNorm(query);
S = movstd(T,[0 m-1]);
for i = 1:n-m+1
    d(i) = sqrt(2*(m-sum(T(i:i+m-1).*Q)/S(i)));
end
```

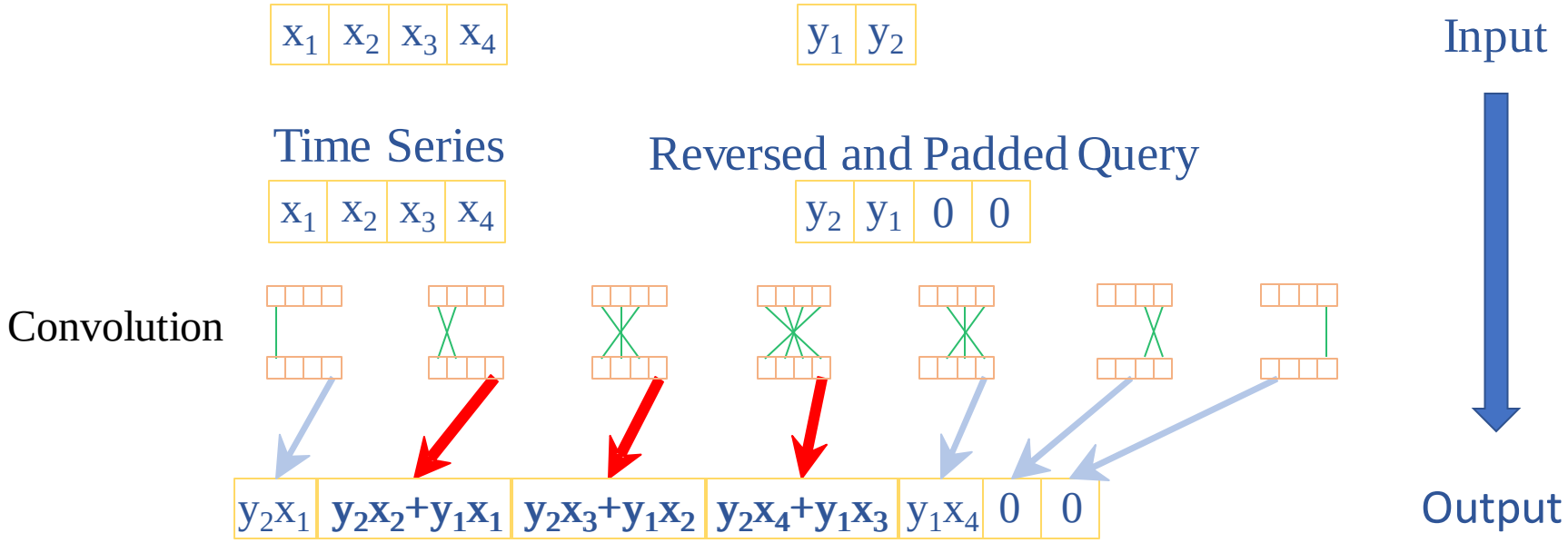
- Still the worst and average cost is $O(nm)$, however, the window is scanned only once per iteration for the dot product.
- Speedup is more than 2X, due to removal of function calls



Thanawin Rakthanmanon, et al. Searching and mining trillions of time series subsequences under dynamic time warping. KDD 2012: 262-270

Mueen's Algorithm for Similarity Search (MASS) (1 of 9)

- Can we improve the just-in-time Normalization algorithm?
- MASS uses a convolution based method to calculate sliding dot products in $O(n \log n)$, in addition to just-in-time z-normalization technique
- **Convolution:** If x and y are vectors of polynomial coefficients, convolving them is equivalent to multiplying the two polynomials.
- We use convolution to compute all of the sliding dot products between the query and sliding windows.



Mueen's Algorithm for Similarity Search (MASS) (2 of 9)

MASS 1.0

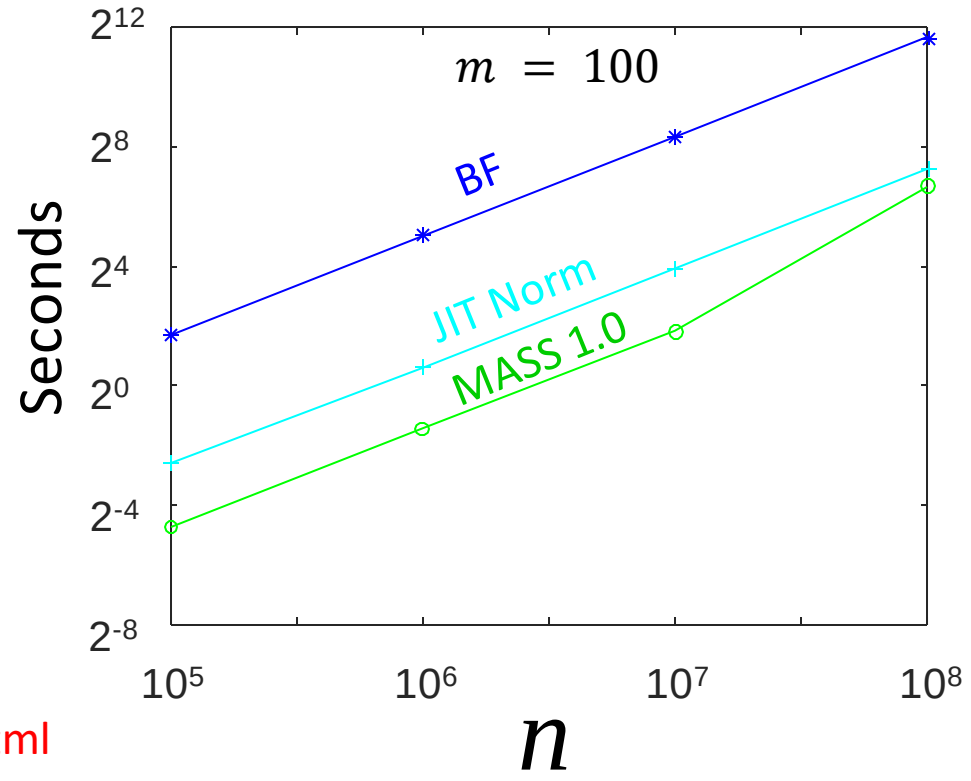
```
d(1:n) = 0;
Q = zNorm(query);
Stdv = movstd(T,[0 m-1]);
Q = Q(end:-1:1);        %Reverse the query
Q(m+1:n) = 0;           %pad zeros
dots = conv(T,Q);
dist = 2*(m-(dots(m:n))./Stdv));
dist = sqrt(dist);
```

The loop has been replaced by the following three lines.

```
d(1:n) = 0;
Q = zNorm(query);
S = movstd(T,[0 m-1]);
for i = 1:n-m+1
    dd(i) = sqrt(2*(m-sum(T(i:i+m-1).*Q)/S(i)));
end
```

Vectorized working formula

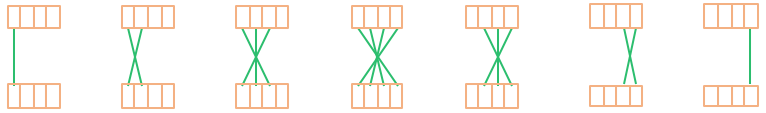
- Computational cost, $O(n \log n)$, does not depend on the query length (m), thus, free of curse of dimensionality.
- There is no loop. Only known mathematical and built-in MATLAB functions are used.

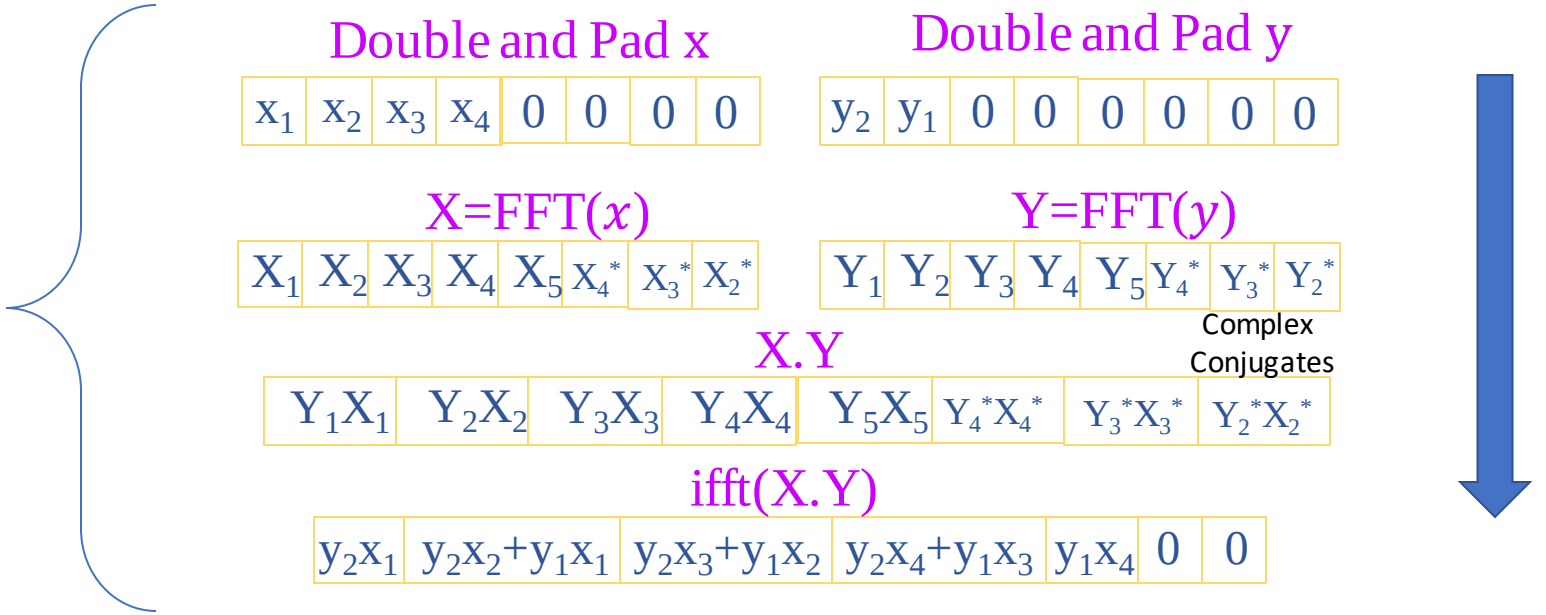


http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html

Mueen's Algorithm for Similarity Search (MASS) (3 of 9)

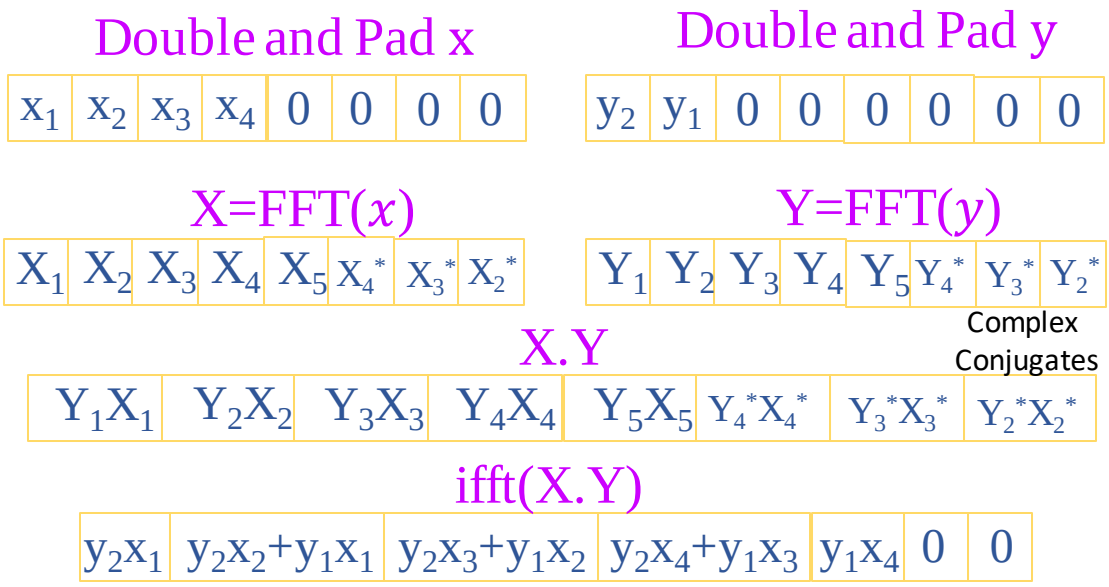
- Can we improve MASS 1.0?
- Note that convolution doubles the size of the input vectors in the output.
- MASS uses only half of the output of convolution and throws away the remaining half.
- Can we compute just the necessary half? Let's see what happens inside convolution.
- *Convolution in time domain is multiplication in frequency domain.*
- `conv(x,y) = ifft( fft( double&pad(x)) . fft(double&pad(y))`

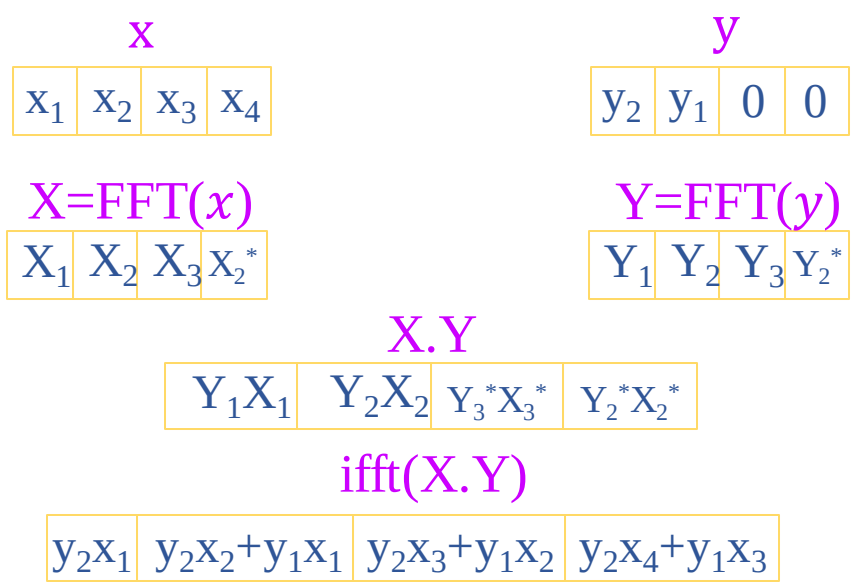




Mueen's Algorithm for Similarity Search (MASS) (4 of 9)

- Can we improve MASS 1.0?
- If we do not double x and y, we obtain a half convolution
- `half conv(x,y)= ifft ( fft(x) . fft(y))`



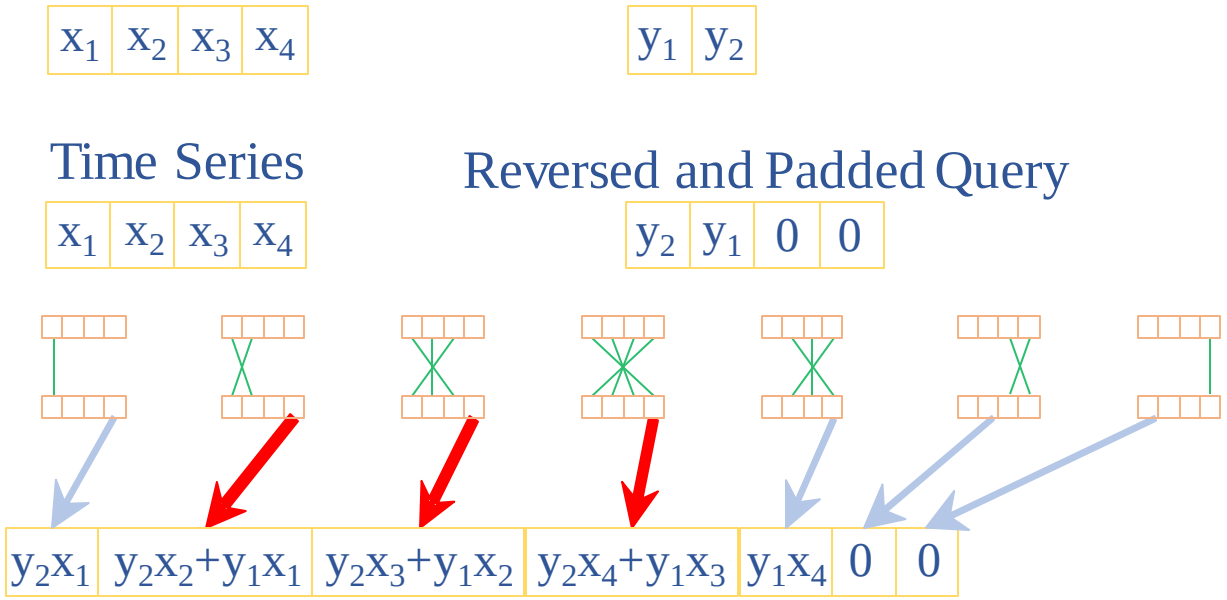


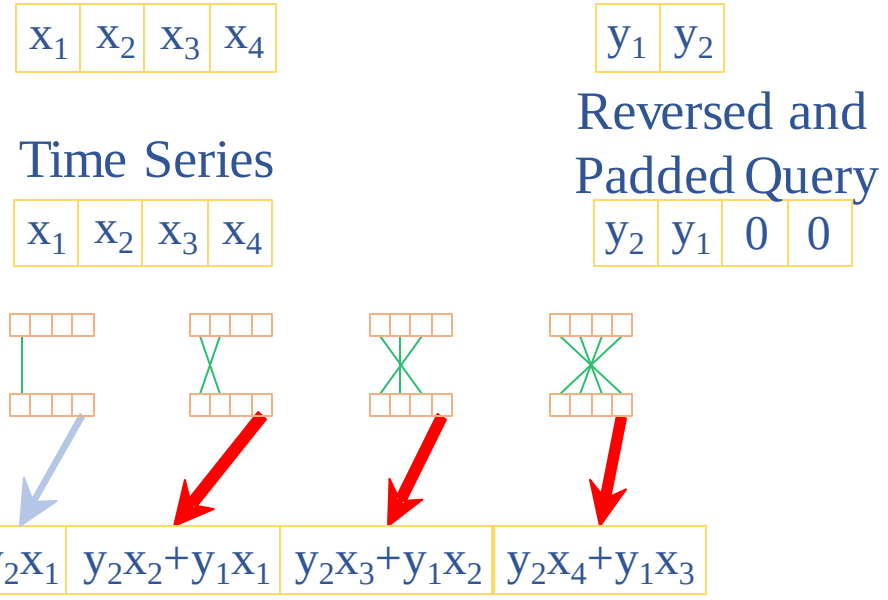

```
conv(x,y) = ifft( fft( double&pad(x)) .
                  fft(double&pad(y) )
```

```
half conv(x,y)= ifft ( fft(x) . fft(y))
```

Mueen's Algorithm for Similarity Search (MASS) (5 of 9)

- Can we improve MASS 1.0?
- Half convolution adds a constraint, $n > \frac{m}{2}$. The constraint is not limiting because the original assumption is $n \gg m$.






```
conv(x,y) = ifft( fft( double&pad(x)) .
                  fft(double&pad(y) )
```

```
half conv(x,y)= ifft ( fft(x) . fft(y))
```

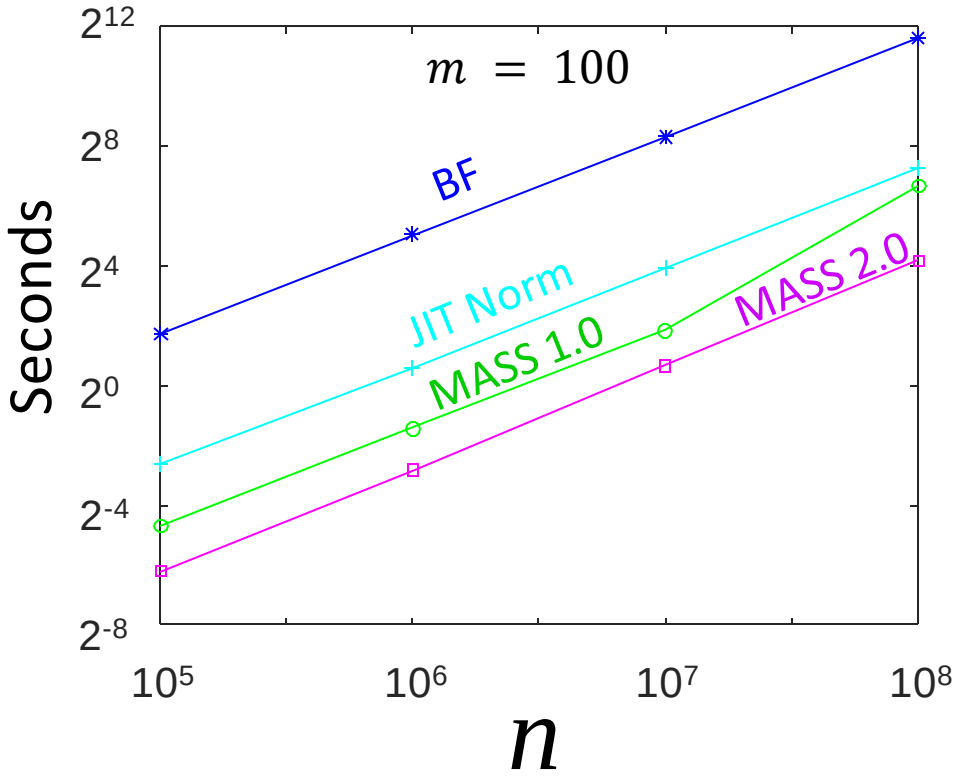
Mueen's Algorithm for Similarity Search (MASS) (6 of 9)

MASS 2.0

```
d(1:n) = 0;
Q = zNorm(query);
Stdv = movstd(T,[0 m-1]);
Q = Q(end:-1:1);        %Reverse the query
Q(m+1:n) = 0;           %pad zeros
dots = ifft( fft(T).* fft(Q) );
dist = 2*(m-(dots(m:n))./Stdv));
dist = sqrt(dist);
```

The `conv(T,Q)` has been replaced, no doubling of sizes

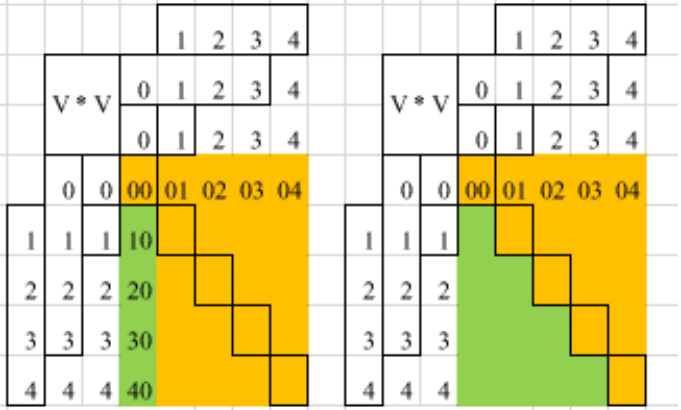
- Computational cost is still $O(n \log n)$, does not depend on the query length (m), thus, free of curse of dimensionality.
- fast Fourier transform (fft) is used as a subroutine



STOMP vs SCRIMP

생략하지 않고 계산

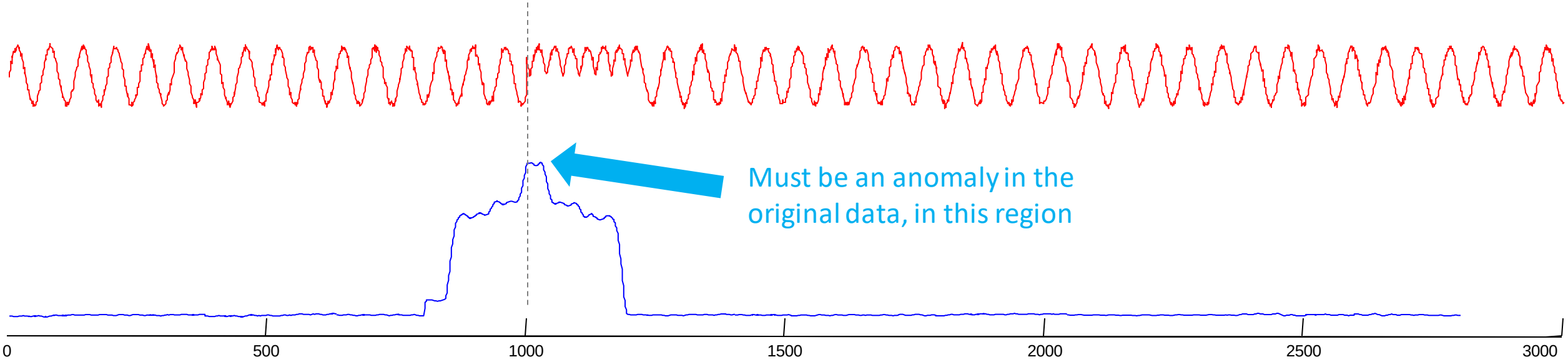
대칭->중복계산->생략



thr0 pi0 = 00 01 02 03 04	thr0 pi0 = 00 01 02 03 04
thr0 pi1 = xx 11 12 13 14	thr0 pi0 = xx 11 12 13 14
thr1 pi2 = xx 21 22 23 24	thr0 pi0 = xx xx 22 23 24
thr2 pi3 = xx 31 32 33 34	thr0 pi0 = xx xx xx 33 34
thr3 pi4 = xx 41 42 43 44	thr0 pi0 = xx xx xx xx 44
총 연산량은 25개	
연산량은 21개, 걸린시간 9개	연산량은 15개, 걸린시간 15개

How to "read" a Matrix Profile: Synthetic Anomaly Example

Where you see relatively high values, you know that the subsequence in the original time series must be unique in its shape. In fact, the highest point is *exactly* the definition of Time Series Discord, perhaps the best anomaly detector for time series*

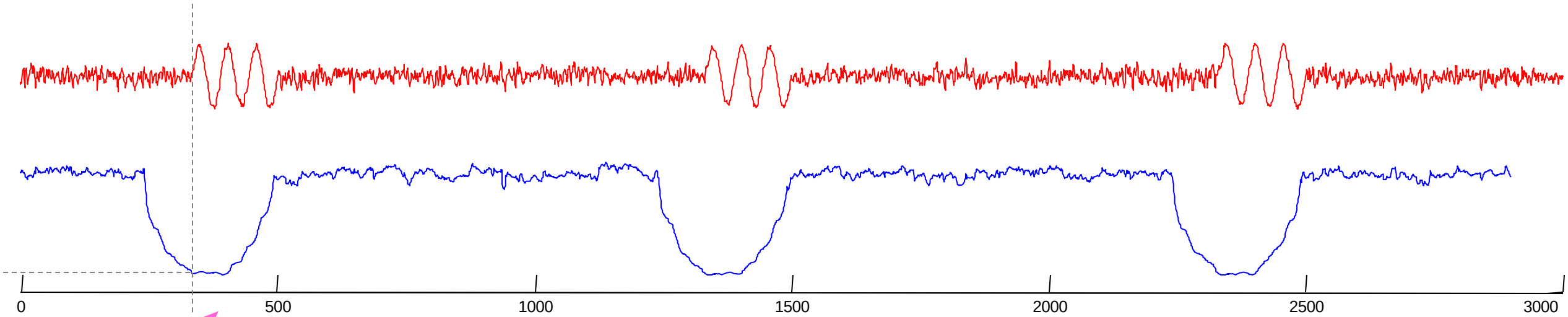


* Vipin Kumar performed an extensive empirical evaluation and noted that "..on 19 different publicly available data sets, comparing 9 different techniques (time series discords) is the best overall technique.". V. Chandola, D. Cheboli, V. Kumar. Detecting Anomalies in a Time Series Database. UMN TR09-004

How to "read" a Matrix Profile: Synthetic Motif Example

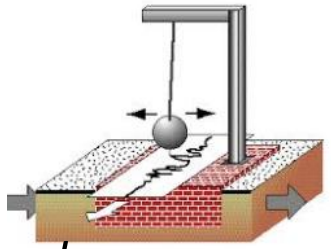
Where you see relatively low values, you know that the subsequence in the original time series must have (at least one) relatively similar subsequence elsewhere in the data.

In fact, the lowest points must be a tieing pair, and correspond exactly to the classic definition of *time series motifs*.

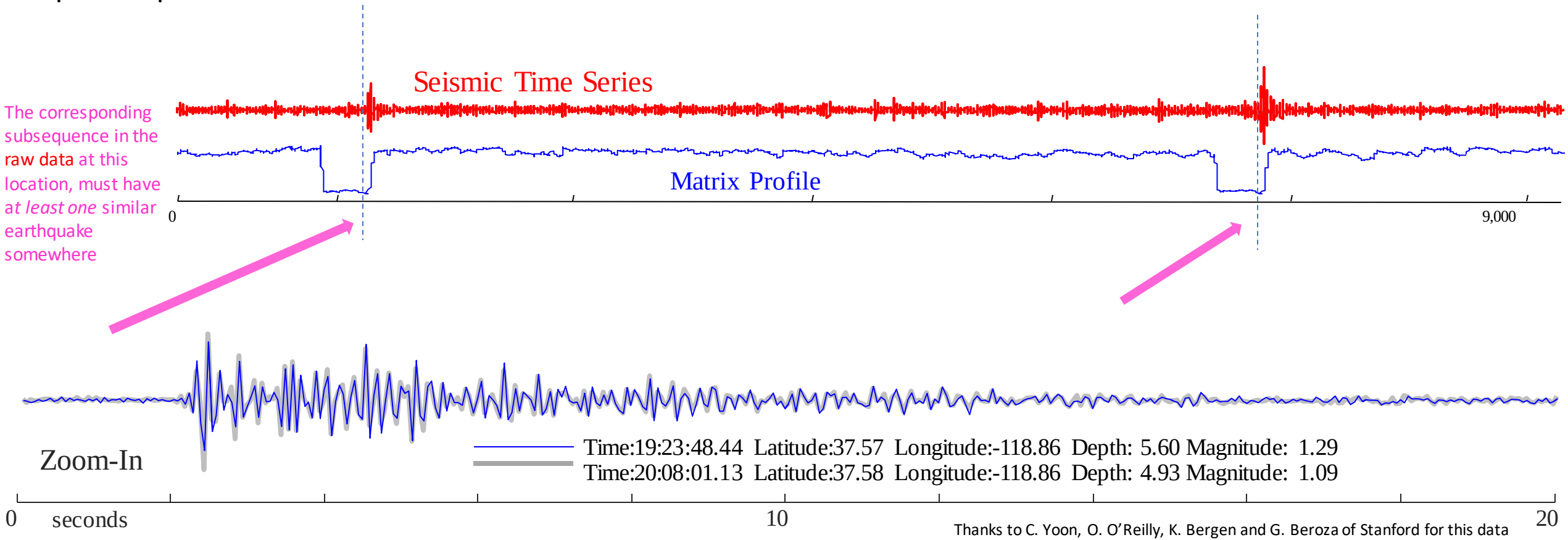


The corresponding subsequence in the raw data at this location, must have a*t least one* similar subsequence somewhere

Real Example 1 : Seismology



If we see low values in the MP of a seismograph, it means there must have been a *repeated earthquake*.
Repeated earthquakes can happen *decades* apart.
Many fundamental problems seismology, including the discovery of foreshocks, aftershocks, triggered earthquakes, swarms, volcanic activity and induced seismicity, can be reduced to the discovery of these repeated patterns.



Thanks to C. Yoon, O. O'Reilly, K. Bergen and G. Beroza of Stanford for this data

Real Example 2 : New York City Taxi – Abnormal Detection

https://www.kaggle.com/c/new-york-city-taxi-fare-prediction/kernels
https://www.kaggle.com/breemen/nyc-taxi-fare-data-exploration
https://github.com/wooheaven/Python-Study/blob/master/06_MatrixProfile/02_matrixprofile-ts/02_Matrix_Profile_NYC_Taxi.ipynb
https://aws.amazon.com/ko/blogs/korea/use-the-built-in-amazon-sagemaker-random-cut-forest-algorithm-for-anomaly-detection/