SAX: A Novel Symbolic Representation of Time Series

A collaboration project between Rorodata and Aayush Kumar for an internship.

Prepared by : Aayush Kumar
Email : aayushkumarjvs@gmail.com

First off, I am grateful to Dr. Ananth Krishnamoorthy, for giving me the opportunity to work with Rorodata and constantly checking in on my progress and giving me advice and tips, for this project.

I would also like to thank people at rorodata for helping me through this project which so that students like me learn to contribute to some, very prestigious open source projects.

Last but not least, I would like to thank my friends, family and fellow hostel mates for their constant support. Also, all the people who have indirectly played a role in my project.

Abstract

Predictive analysis on Time Series Data Sets using SAX: The potential to do predictive machine learning on the data generated by connected sensors (temperature sensor, LIDAR etc) is a feature that is driving the spread of the Internet of Things. Predictive analysis on the time series data can be used to anticipate adverse events, enable early warning systems, improve results, reduce costs and enhance efficiency.

An Anomaly (a.k.a outlier) is simply an unusual subsequence of the series. “Unusual” can be taken as “improbable”. Probability is not defined for the time series, Probability can be defined for Symbols. Mapping a time series to a symbol may allow us to assign a probability to the time series subsequence. This involves mapping the time series subsequence to a symbol in some symbol space. Key advantages of using SAX is that it yields an explainable model. Wherein the result of the model should not be underestimated.

Table of Contents

Internship Workflow

1.Data Collection and Interpreting: After getting data from rorodata Using tools like Feather (binary file format for non-csv), ParaText (helps in a parallel reading of csv) the data will be used for understanding patterns that it exhibit and interpret them.

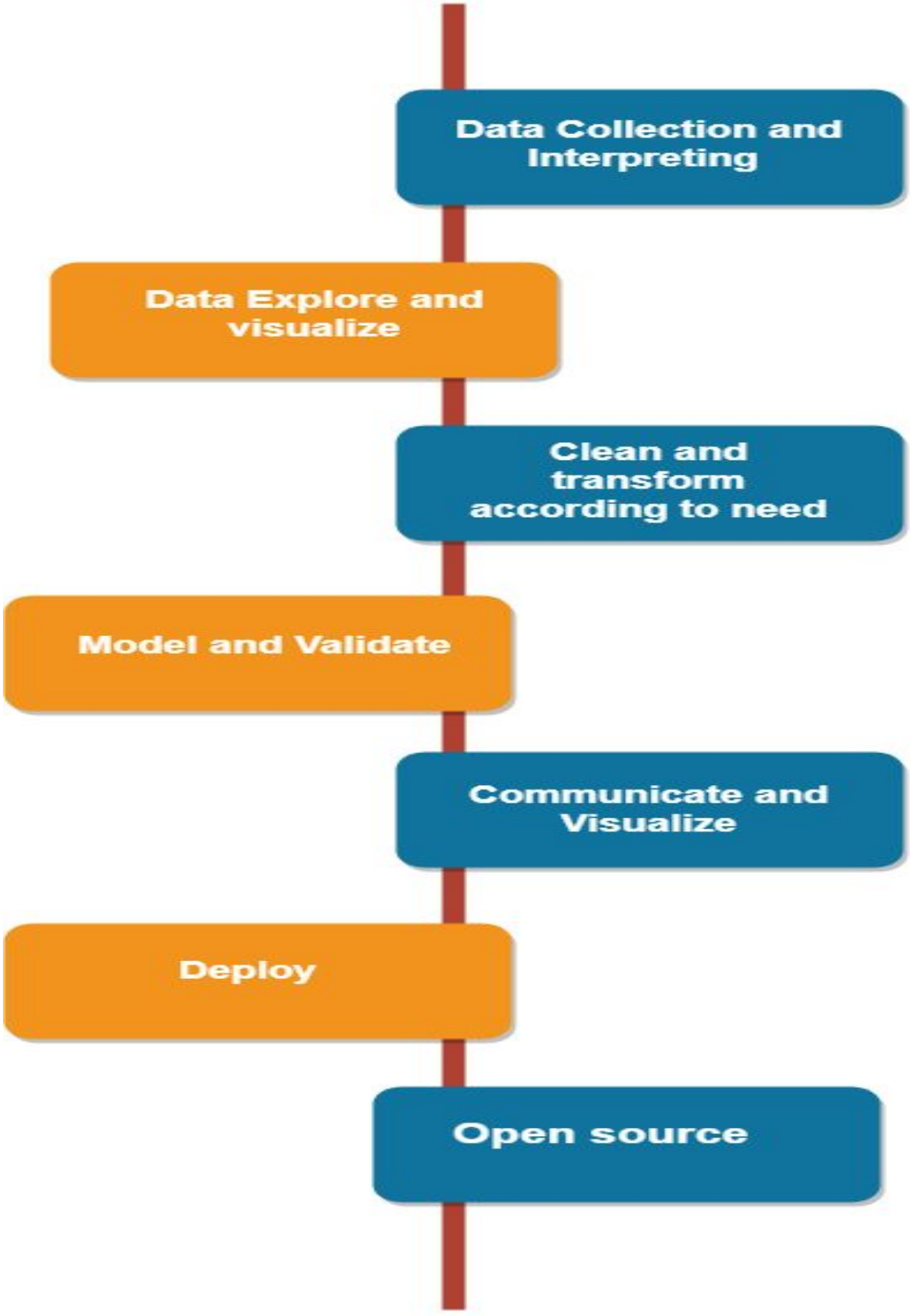
2. Data Explore and Visualize: Data is then used to detect and visualize patterns from tools like Seaborn(matplotlib based statistic visualizer), Bokeh(interactive/d3.js like), Plotly(declarative data visualizer), VizTree and Geoplotlib (for maps based data).

3.Clean and Transform: Once data is obtained and visualized spending time on cleaning and transforming according to our need on the basis of predictive analysis. data cleaner(automate cleaning your data in Pandas), Blaze(Numpy/Pandas), xarray(pandas support).

4.Model and Validate: Keras(Tensorflow), Stat Modeling based on obtained data mostly inclined toward Symbolic Aggregate ApproXimate (SAX)

5. Communicate and Visualize: Discuss the validation of the performance of algorithm over data, visualization and change hyper-parameter according to the need and performance.

6.Deploy & Open-Source: If the model turns using rorodata platforms one-click deployment feature for the people who can use the algorithm for their sensor data.

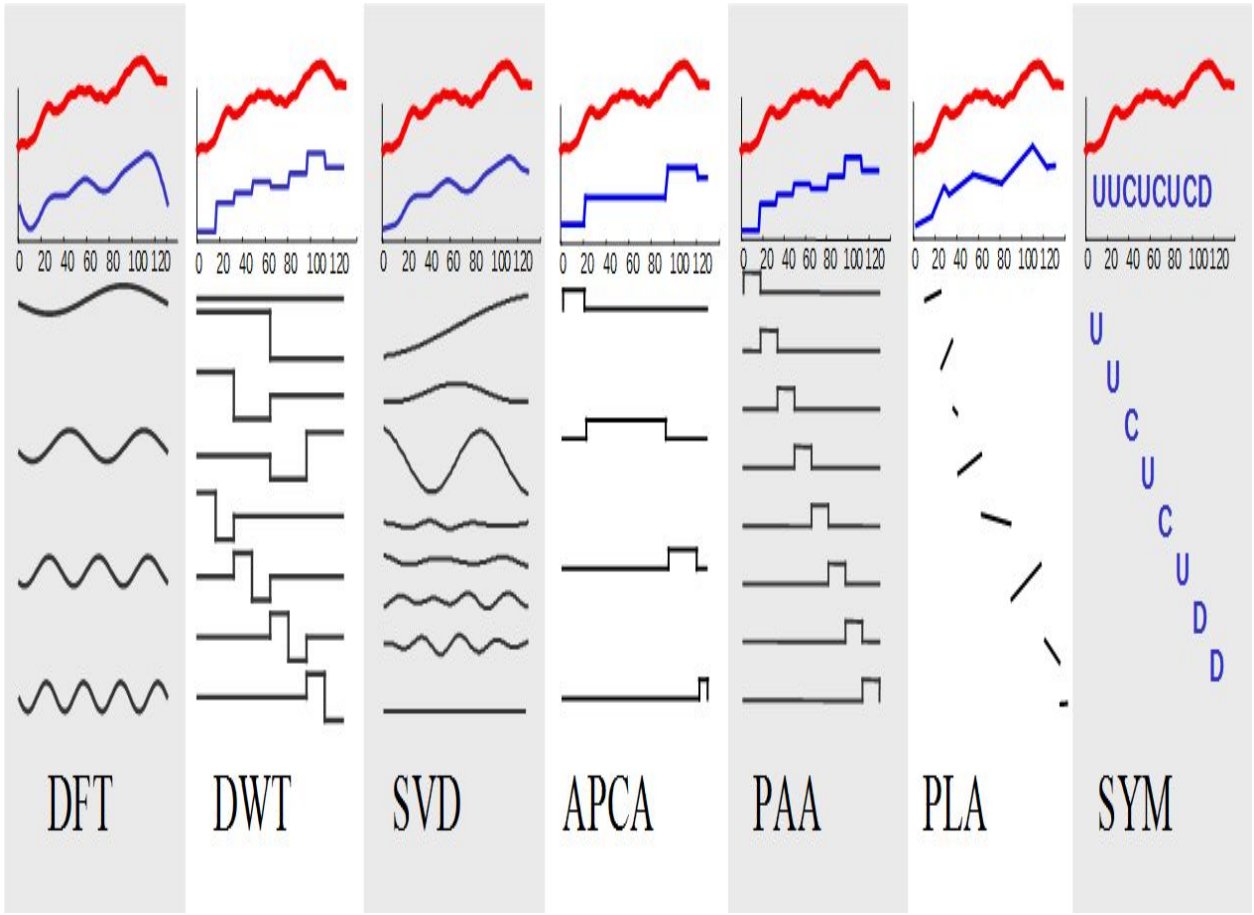

Data Collection and Interpreting
Data Explore and visualize
Clean and transform according to need
Model and Validate
Communicate and Visualize
Deploy
Open source

A time series is a sequence of pairs where each pair consists of a time index and value. The time index may be implied if there is a constant difference between values.

The time series can be segmented into “Windows” which represent the time series between two-time indices.

The symbol can represent Windows Because symbols in a finite symbol space and have a probability.

Symbols are easy to store and manipulate - each symbol can be represented by an integer or a float variable. Creating a few approximation techniques helps to fit the data in main memory.



Symbolic representation of Time Series dataset

Advantages of symbolic representation are

1. Lower bounding of Euclidean Distance
2. Dimensionality Reduction
3. Numerosity Reduction

This representation is known as SAX Symbolic Aggregate Approximation.

What is SAX?

SAX is a good representation of working in raw data for most problems and representing time series data in the form of strings with a fixed length size.

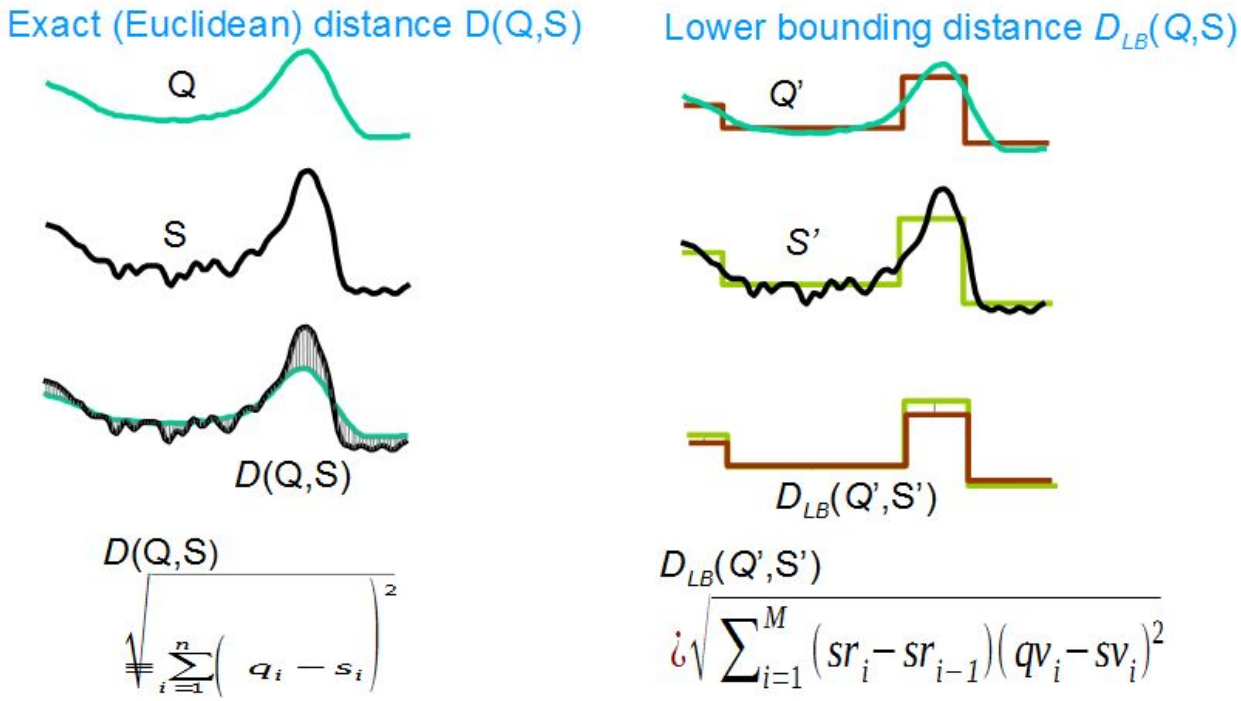
SAX is a methodology for reducing a time series window to a symbols

The technique was developed by Dr Eammonn Keogh et al UC Riverside in the early 2000s.

It has drawn a great deal of attention in a world of time series analysis. Allows a time series of arbitrary length n to be reduced to a string of arbitrary length w ($w<n$).

SAX is the first symbolic representation for time series that allows for dimensionality reduction and indexing with a lower-bounding distance measure.

What is lower bounding?



Lower bounding means that for all Q and S, we have…

$$D_{LB}(Q',S') \le D(Q,S)$$

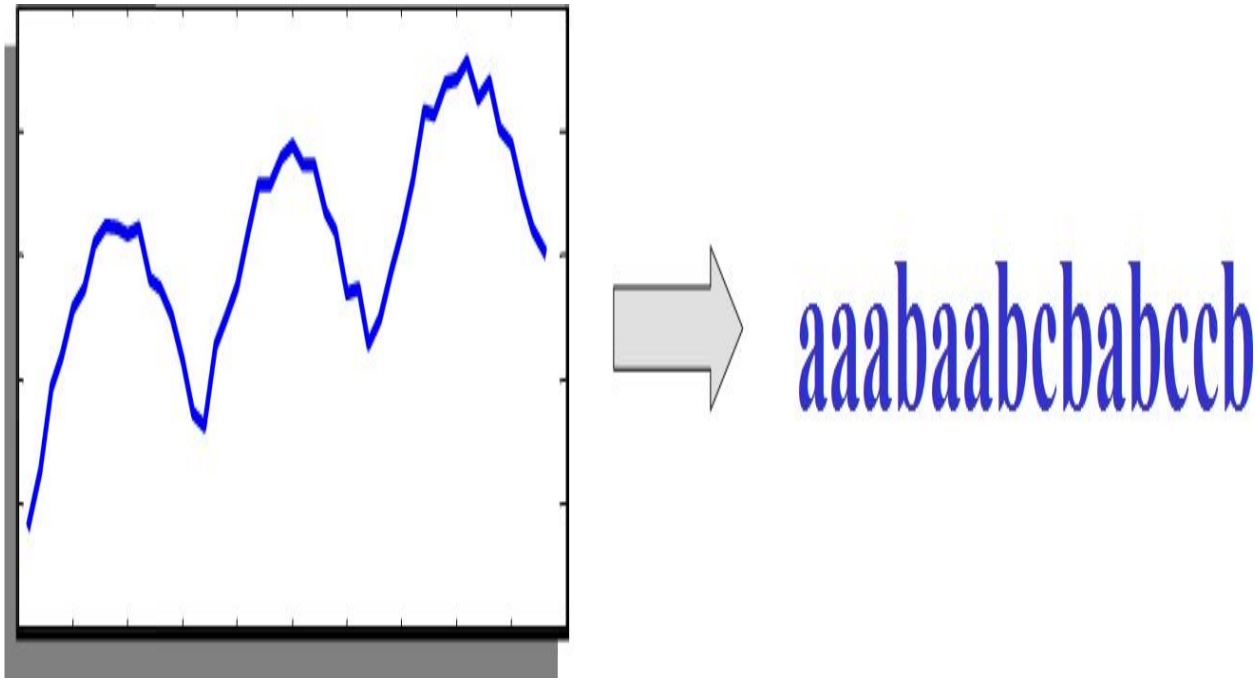
What's a SAX word?

A SAX word is the symbol generated by the SAX algorithm. It is defined by a SAX alphabet and a length. The SAX is traditionally represented by letters, and its components are referred to as "SAX letters"

The size of the alphabet is typically small - this is important for anomaly detection

When we write out a description of a SAX word, we typically use a string like representation, such as "abcdefg"

SAX letters don't have to be letters - implementations often use numbers based at zero, although we display them as letters.

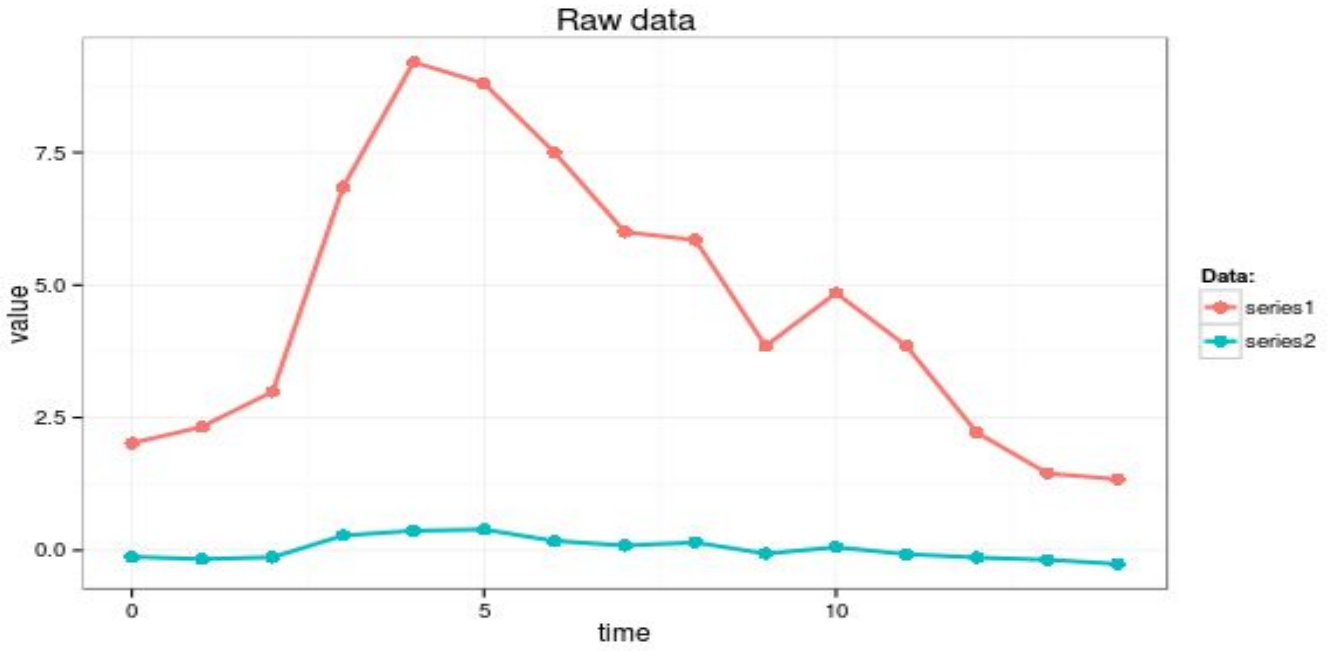


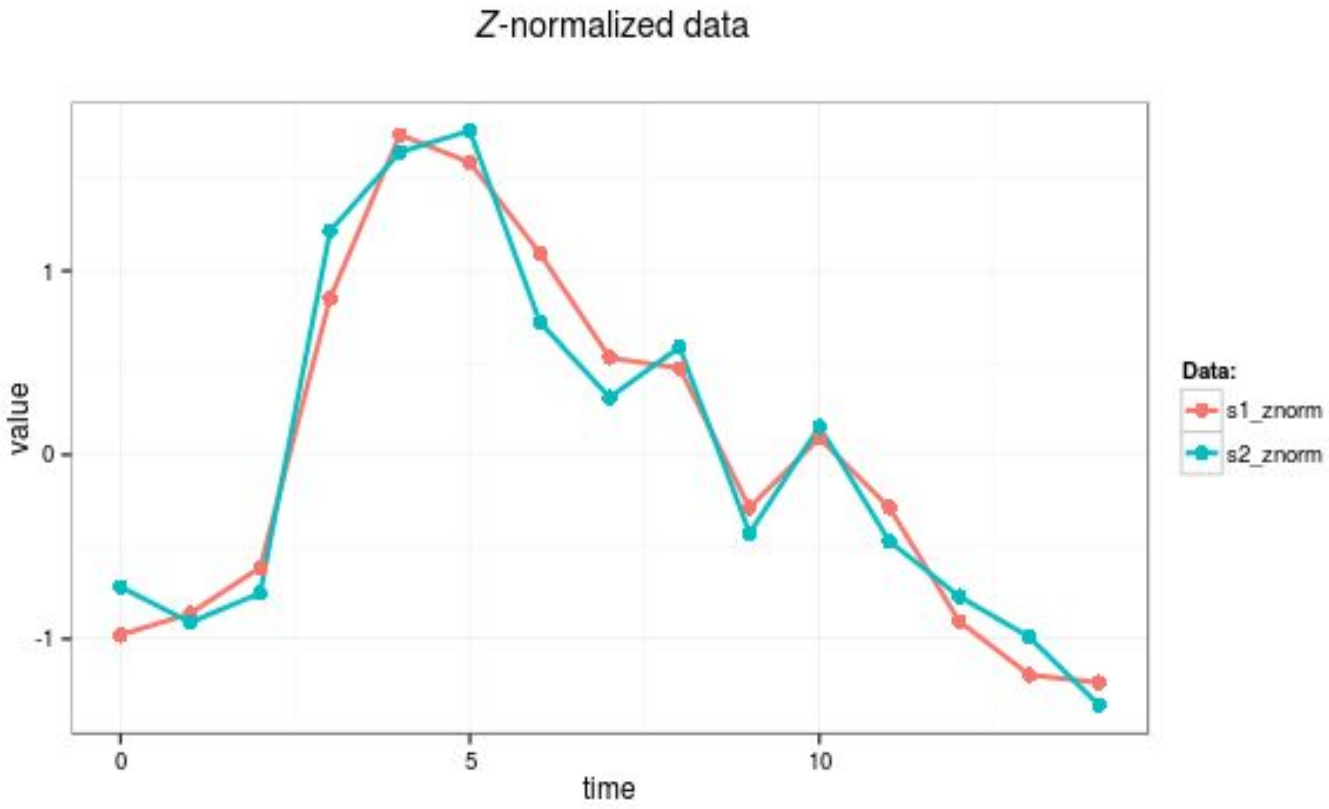
Normalization of Time Series

Normalization to zero mean and unit of energy

The procedure ensures, that all elements of the input vector are transformed into output vector whose mean is approximately 0 while the standard deviation is in a range close to 1, The formula behind transform is shown here:

$x'i=(xi-\mu)/\sigma$, where $i \in N$


Raw data
7.5
5.0
2.5
0.0
value
0
5
10
time
Data:
series1
series2


Z-normalized data
1
0
-1
value
0
5
10
time
Data:
s1_znorm
s2_znorm

How to obtain SAX?

First convert the time series to PAA representation, then convert the PAA to symbols, It takes linear time.

Data is divided into w equal-sized frames

A mean value of the data falling within the frame is calculated

Vectors of these values become PAA

Reduce dimensions by PAA

PAA approximates a time-series *X* of length *n* into vector $X = (\bar{x}_1, \ldots, \bar{x}_M)$ of any arbitrary length $M \leq n$ where each of $\bar{x}_i$ is calculated as follows:

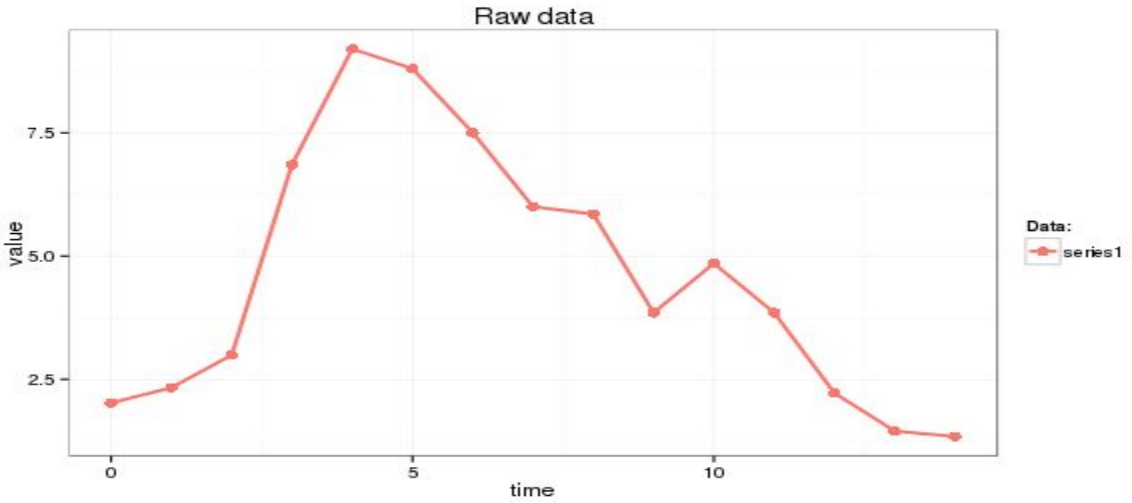
$$\bar{x}_i = \frac{M}{n} \sum_{j=n/M(i-1)+1}^{(n/M)i} x_j$$

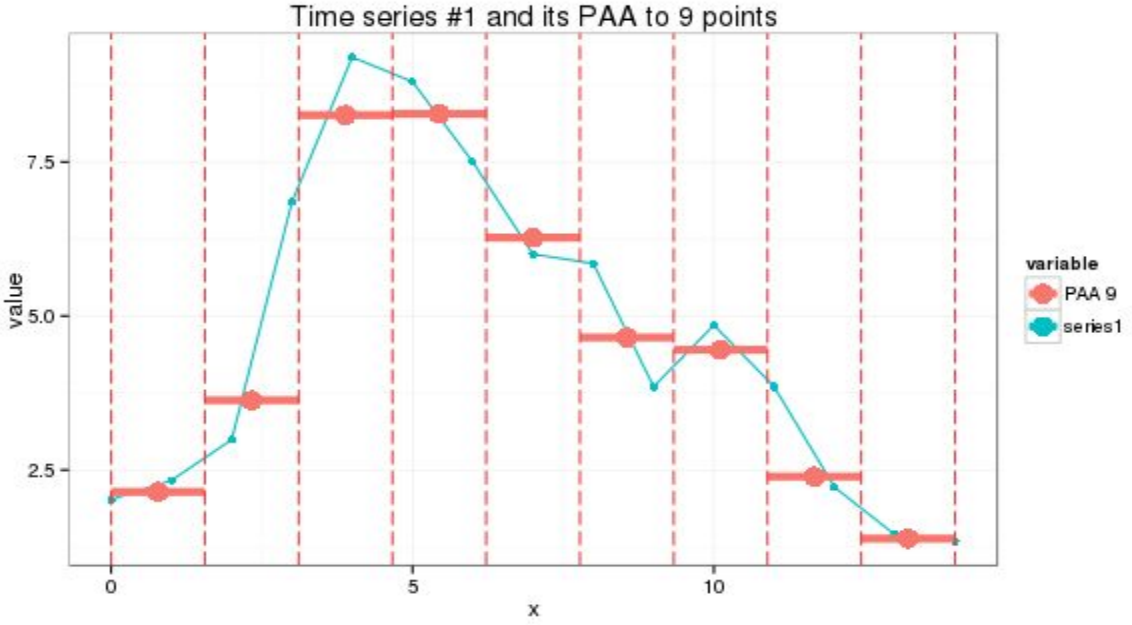
Which simply means that in order to reduce the dimensionality from *n* to *M*, we first divide the original time-series into *M* equally sized frames and secondly compute the mean values for each frame. The sequence assembled from the mean values is the PAA approximation (i.e., transform) of the original time-series. As it was shown by Keogh et al, the complexity of the PAA transform can be reduced from *O(NM)* to *O(Mm)* where *m* is the number of frames. By using the following distance measure

$$D_{PAA}(\bar{X}, \bar{Y}) \equiv \sqrt{\frac{n}{M}} \sqrt{\sum_{i=1}^{M} (\bar{x}_i - \bar{y}_i)}$$

Yi & Faloutsos, and Keogh et al, have shown that PAA satisfies to the lower bounding condition and guarantees no false dismissals, i.e.:

$$D_{PAA}(\bar{X}, \bar{Y}) \leq D(X, Y)$$


Raw data
value
7.5
5.0
2.5
0
5
10
time
Data:
series1

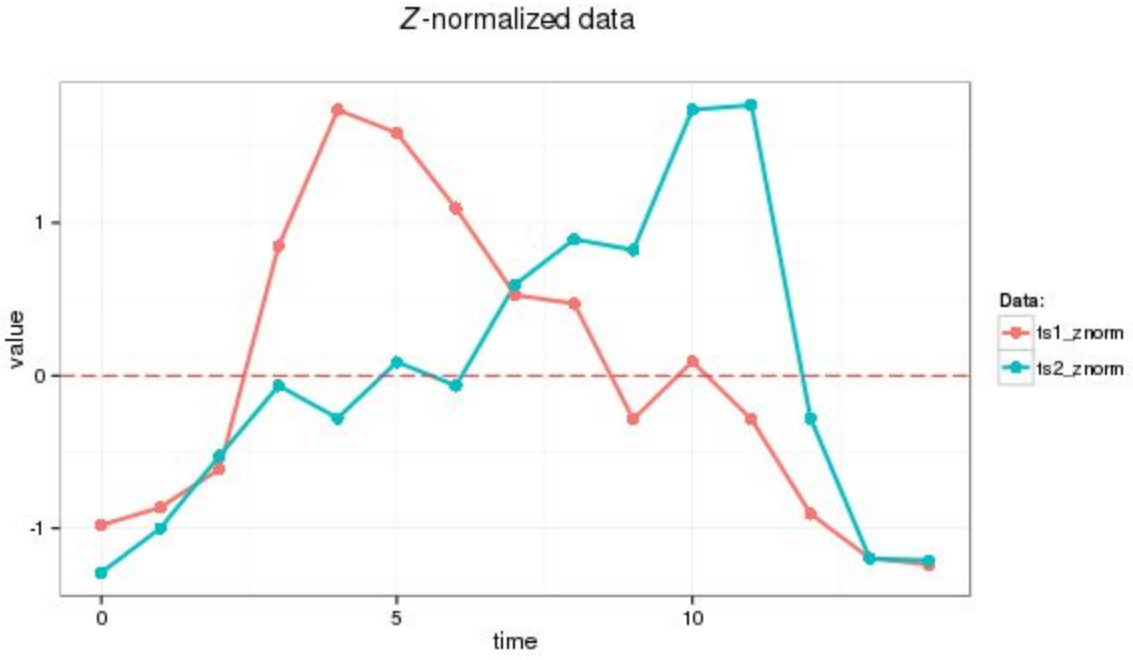

Time series #1 and its PAA to 9 points
value
7.5
5.0
2.5
0
5
10
x
variable
PAA 9
series1

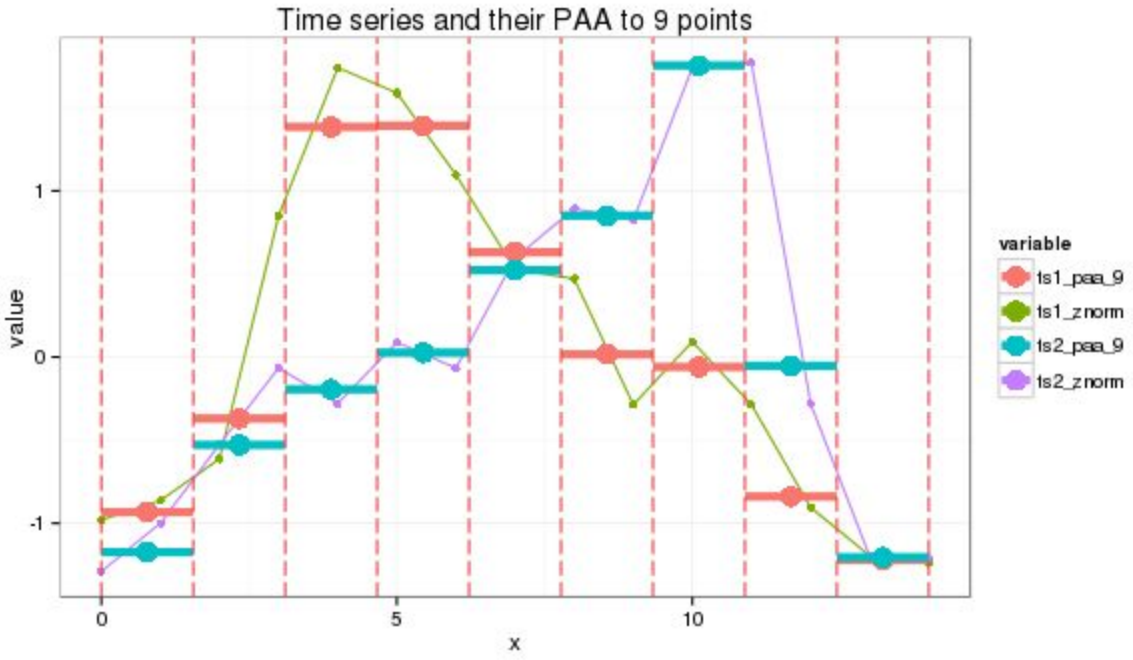
SAX transforms a time-series *X* of length *n* into the string of arbitrary length ω, where ω«nω«n typically, using an alphabet *A* of size *a* > *2*. The algorithm consists of two steps: (i) it transforms the original time-series into the PAA representation and (ii) it converts the PAA data into a string.

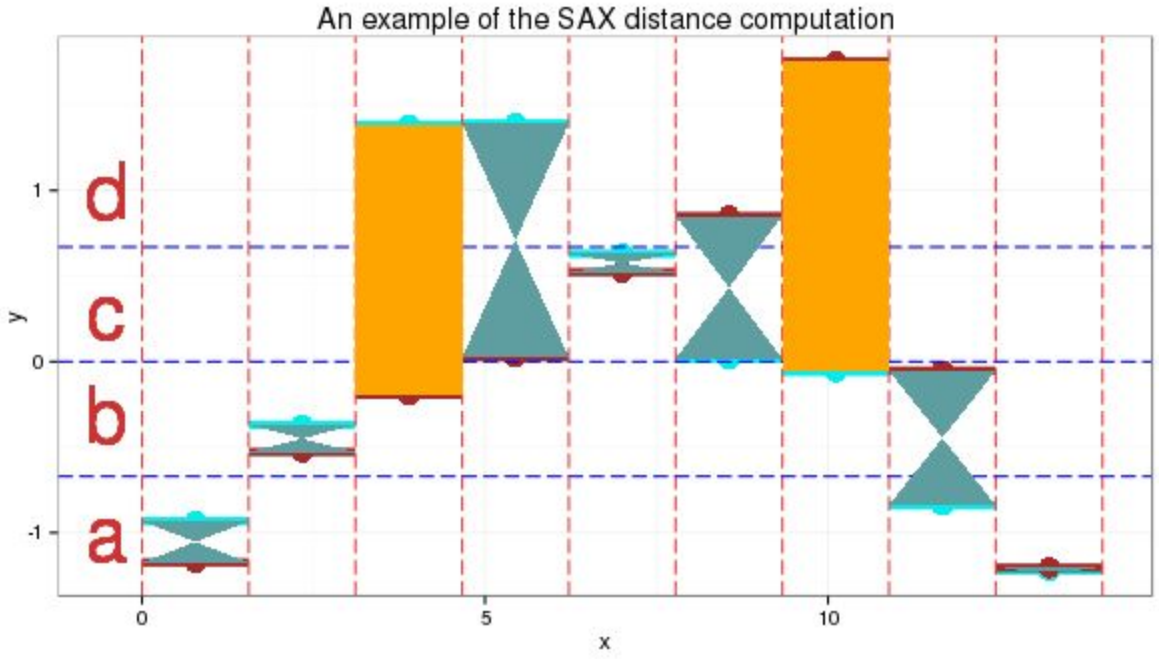
$$MINDIST(\hat{Q},\hat{C}) \equiv \sqrt{\frac{n}{w}}\sqrt{\sum_{i=1}^{w}(dist(\hat{q}*i,\hat{c}*i))^2}$$

As shown by Li et al, this SAX distance metrics lower-bounds the PAA distance, i.e.

$$\sum_{i=1}^{n}(q_i-c_i)^2 \geq n(\bar{Q}-\bar{C})^2 \geq n(dist(\hat{Q},\hat{C}))^2$$


Z-normalized data
value
time
Data:
ts1_znorm
ts2_znorm


Time series and their PAA to 9 points
value
x
variable
ts1_paa_9
ts1_znorm
ts2_paa_9
ts2_znorm



Use the 4 symbols alphabet {a,b,c,d} as in the table above. The cut lines for this alphabet shown as the thin blue lines on the plot below.

SAX transform of ts1 into string through 9-points PAA: “abddccbaa”

SAX transform of ts2 into string through 9-points PAA: “abbccddba”

SAX distance: 0 + 0 + 0.67 + 0 + 0 + 0 + 0.67 + 0 + 0 = 1.34

At the plot, orange color depicts symbols distance between which is counted - they are not “adjacent” to each other in the table.

Gaussian distribution and Distance Measure

Most “natural” distribution. The gaussian process uses lazy learning and a measure of the similarity between points to predict the values for an unseen point from the training data.

Conclusion and Future Work

After getting the data from rorodata stack most of the above Machine learning processes will be continued most of my time will be consumed by predictive analysis of the time series sensor data.

If the work turns out to be notable members of rorodata would help me open-source it, so that it becomes useful for others to learn.

Important Web-links and Code Details

1 https://jmotif.github.io/sax-vsm_site/modules/algorithm/

2 https://www.slideshare.net/goyalnikita277/saxtimeseries

3 https://r2s.hh.se/ReadingClub/2012-09-21/0-sax.pdf

4 http://www.cse.cuhk.edu.hk/~adafu/Pub/icdm05time.pdf

5 https://cs.gmu.edu/~jessica/sax.htm

6 http://www.cs.ucr.edu/~eamonn/SAX.htm

7 http://grammarviz2.github.io/grammarviz2_site/

8 https://github.com/johannfaouzi/pyts

9 http://www.cs.ucr.edu/~eamonn/HOT%20SAX%20%20long-ver.pdf

10 https://github.com/nphoff/saxpy

11 http://alumni.cs.ucr.edu/~ratana/SSDBM05.pdf

12 https://jmotif.github.io/sax-vsm_site/