DATA MINING 2

Time Series – Approximation & Clustering

Riccardo Guidotti

a.a. 2021/2022

Slides edited from Keogh Eamonn's tutorial

UNIVERSITÀ DI PISA

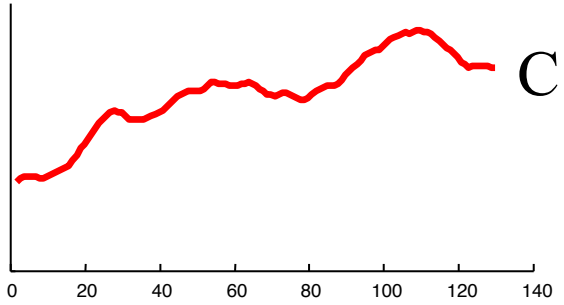
Approximation

Time Series Approximation

- Approximation: represent a TS into a new smaller and simpler space and use this novel representation for computing.
- Approximation is a special form of Dimensionality Reduction specifically designed for TSs.
- Approximation vs Compression: the approximated space is always understandable, while the compressed space is not necessarily understandable.

An Example of an Approximation

- The graphic shows a time series with 128 points.
- The raw data used to produce the graphic is also reproduced as a column of numbers (just the first 30 or so points are shown).



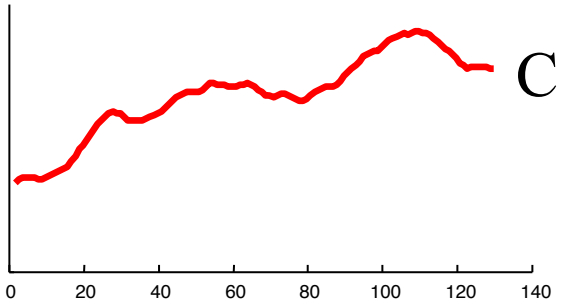
$$n = 128$$

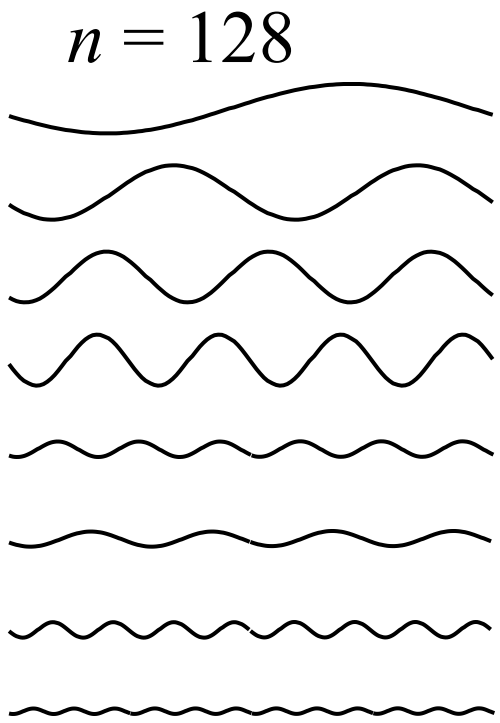
Raw Data

0.4995
0.5264
0.5523
0.5761
0.5973
0.6153
0.6301
0.6420
0.6515
0.6596
0.6672
0.6751
0.6843
0.6954
0.7086
0.7240
0.7412
0.7595
0.7780
0.7956
0.8115
0.8247
0.8345
0.8407
0.8431
0.8423
0.8387
…

An Example of a Approximation

- We can decompose the data into 64 pure sine waves using the Discrete Fourier Transform (just the first few sine waves are shown).
- The Fourier Coefficients are reproduced as a column of numbers (just the first 30 or so coefficients are shown).
- Note that at this stage we have not done dimensionality reduction, we have merely changed the representation...





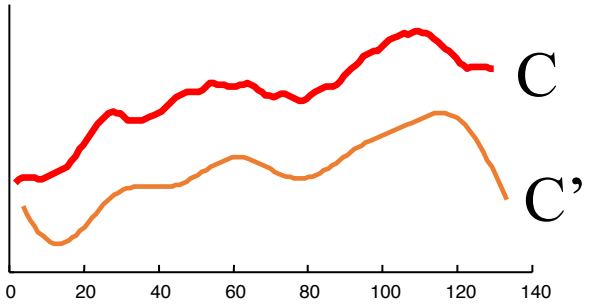
Fourier Coefficients	Raw Data
1.5698	0.4995
1.0485	0.5264
0.7160	0.5523
0.8406	0.5761
0.3709	0.5973
0.4670	0.6153
0.2667	0.6301
0.1928	0.6420
0.1635	0.6515
0.1602	0.6596
0.0992	0.6672
0.1282	0.6751
0.1438	0.6843
0.1416	0.6954
0.1400	0.7086
0.1412	0.7240
0.1530	0.7412
0.0795	0.7595
0.1013	0.7780
0.1150	0.7956
0.1801	0.8115
0.1082	0.8247
0.0812	0.8345
0.0347	0.8407
0.0052	0.8431
0.0017	0.8423
0.0002	0.8387
…	…

An Example of a Approximation

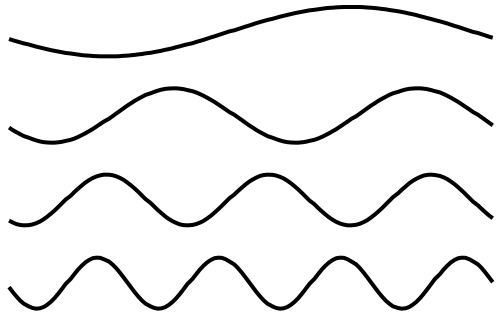
- … however, note that the first few sine waves tend to be the largest (equivalently, the magnitude of the Fourier coefficients tend to decrease as you move down the column).
- We can therefore truncate most of the small coefficients with little effect.

$n = 128$
$N = 8$
$C_{\text{ratio}} = 1/16$

We have discarded $\frac{15}{16}$ of the data.



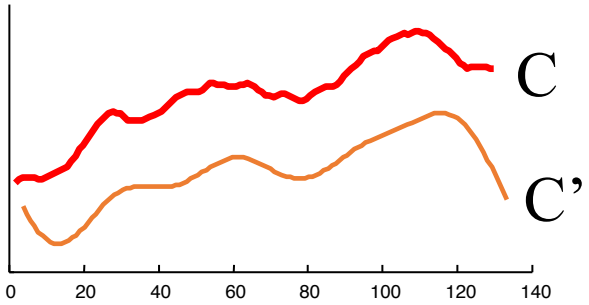
$n = 128$

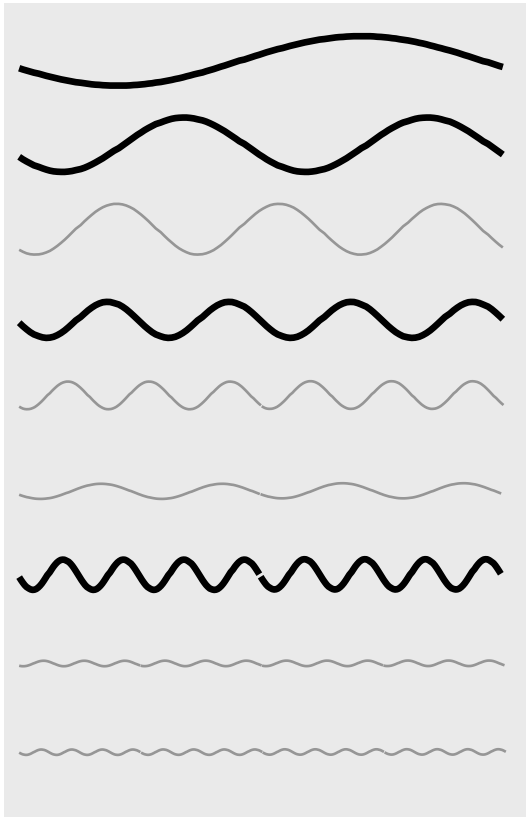


Truncated Fourier Coefficients	Fourier Coefficients	Raw Data
1.5698	1.5698	0.4995
1.0485	1.0485	0.5264
0.7160	0.7160	0.5523
0.8406	0.8406	0.5761
0.3709	0.3709	0.5973
0.4670	0.4670	0.6153
0.2667	0.2667	0.6301
0.1928	0.1928	0.6420
	0.1635	0.6515
	0.1602	0.6596
	0.0992	0.6672
	0.1282	0.6751
	0.1438	0.6843
	0.1416	0.6954
	0.1400	0.7086
	0.1412	0.7240
	0.1530	0.7412
	0.0795	0.7595
	0.1013	0.7780
	0.1150	0.7956
	0.1801	0.8115
	0.1082	0.8247
	0.0812	0.8345
	0.0347	0.8407
	0.0052	0.8431
	0.0017	0.8423
	0.0002	0.8387
	…	…

An Example of a Approximation

- Instead of taking the first few coefficients, we could take the best coefficients
- This can help greatly in terms of approximation quality, but makes indexing hard (impossible?).
- Note this applies also to Wavelets





Truncated Fourier Coefficients	Fourier Coefficients	Raw Data
1.5698	1.5698	0.4995
1.0485	1.0485	0.5264
0.7160	0.7160	0.5523
0.8406	0.8406	0.5761
0.3709	0.3709	0.5973
0.4670	0.4670	0.6153
0.2667	0.2667	0.6301
0.1928	0.1928	0.6420
	0.1635	0.6515
	0.1602	0.6596
	0.0992	0.6672
	0.1282	0.6751
	0.1438	0.6843
	0.1416	0.6954
	0.1400	0.7086
	0.1412	0.7240
	0.1530	0.7412
	0.0795	0.7595
	0.1013	0.7780
	0.1150	0.7956
	0.1801	0.8115
	0.1082	0.8247
	0.0812	0.8345
	0.0347	0.8407
	0.0052	0.8431
	0.0017	0.8423
	0.0002	0.8387
	...	...

Agrawal, Faloutsos, & Swami. FODO 1993

Faloutsos, Ranganathan, & Manolopoulos. SIGMOD 1994

Chan & Fu. ICDE 1999

Korn, Jagadish & Faloutsos. SIGMOD 1997

Keogh, Chakrabarti, Pazzani & Mehrotra SIGMOD 2001

Keogh, Chakrabarti, Pazzani & Mehrotra KAIS 2000

Yi & Faloutsos VLDB 2000

Morinaka, Yoshikawa, Amagasa, & Uemura, PAKDD 2001

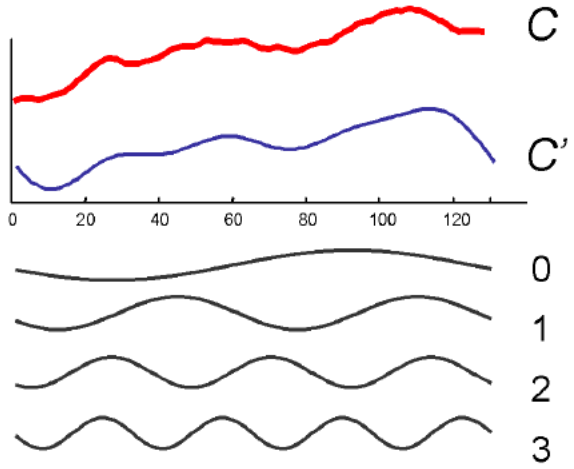
Lin, J., Keogh, E., Lonardi, S. & Chiu, B. (2003) A Symbolic ACM SIGMOD Workshop

Discrete Fourier Transform (DFT)

Jean Fourier

1768-1830

- Represent the time series as a linear combination of sines and cosines, but keep only the first n/2 coefficients.
- Why n/2 coefficients? Because each sine wave requires 2 numbers, for the phase (w) and amplitude (A,B).
- A TS represented with DTF is said to be in the frequency domain.
- Many of the Fourier coefficients have very low amplitude and thus contribute little to reconstructed signal. These low amplitude coefficients can be discarded without much loss of information thereby saving storage space.
- Pros
  - Good ability to compress most natural signals.
  - Fast, off the shelf DFT algorithms exist. O(nlog(n)).
- Cons
  - Difficult to deal with sequences of different lengths.
  - Cannot support weighted distance measures.

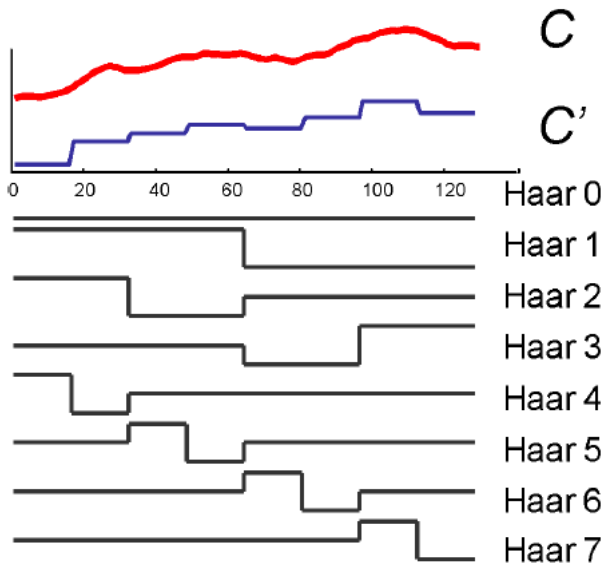


Discrete Wavelet Transform (DWT)

Alfred Haar

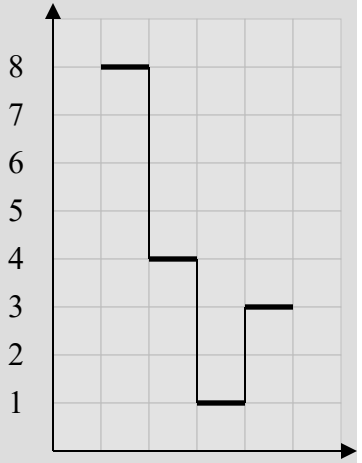
1885-1933

- Represent the TS as a linear combination of Wavelet basis functions, but keep only the first N coefficients.
- Wavelets are represented data in terms of the sum and difference of a prototype function, so called the “analyzing” or “mother” wavelet.
- Wavelets are localized in time, i.e., some of the ***wavelet*** coefficients represent small, ***local*** subsections of the data. This is in contrast to ***Fourier*** coefficients that always represent ***global*** contribution to the data.
- Haar wavelets seem to be as powerful as the other wavelets for most problems and are very easy to code.
- Pros
  - Good ability to compress stationary signals.
  - Fast linear time algorithms for DWT exist.
- Cons
  - It is only defined for sequence whose length is an integral power of two.
  - Otherwise wavelets approximate the left side of signal at the expense of the right side.
  - Cannot support weighted distance measures.

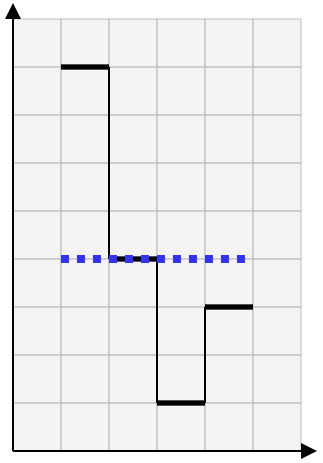


Not available in Python

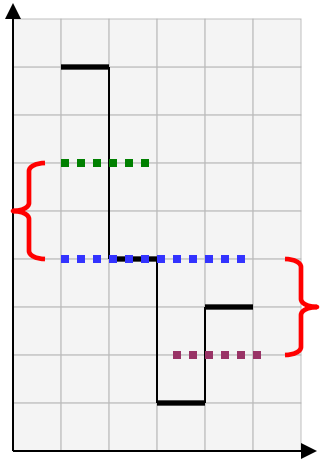
X = {8, 4, 1, 3}



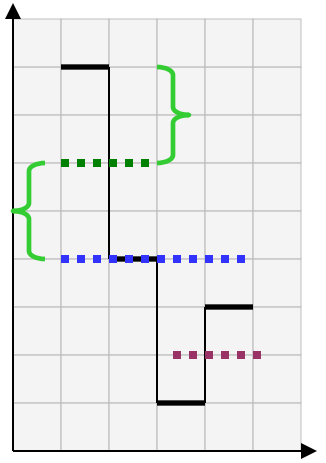
h_1 = 4 = mean(8,4,1,3)



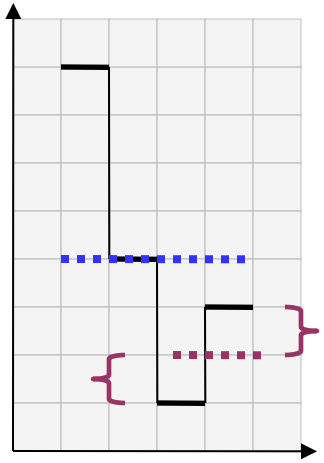
h_2 = 2 = mean(8,4) - h_1



h_3 = 2 = (8-4)/2

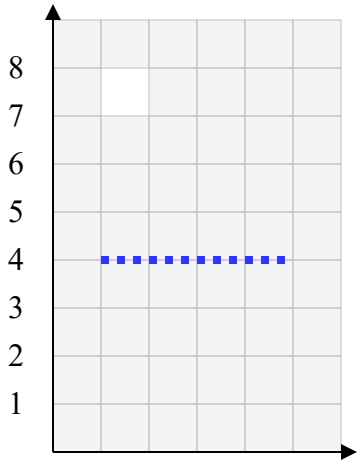


h_4 = -1 = (1-3)/2

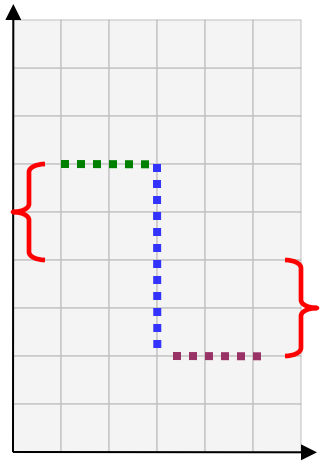


I have converted a raw time series X = {8, 4, 1, 3}, into the Haar Wavelet representation H = [4, 2 , 2, -1]
We can covert the Haar representation back to raw signal with no loss of information...

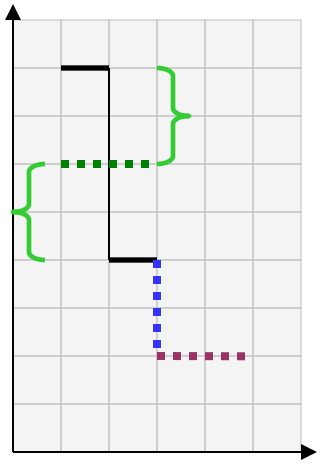
h_1 = 4



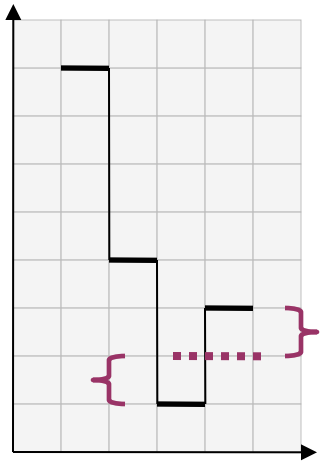
h_2 = 2



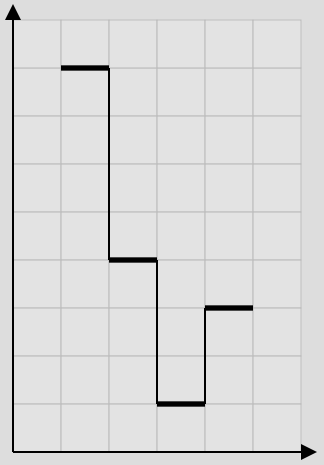
h_3 = 2



h_4 = -1



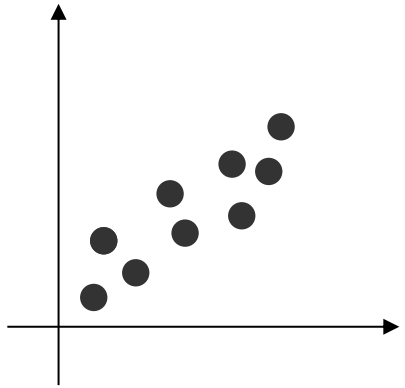
X = {8, 4, 1, 3}

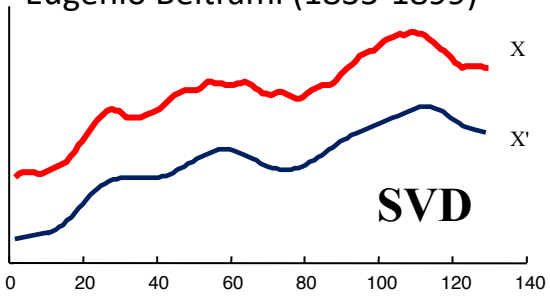


Singular Value Decomposition (SVD)

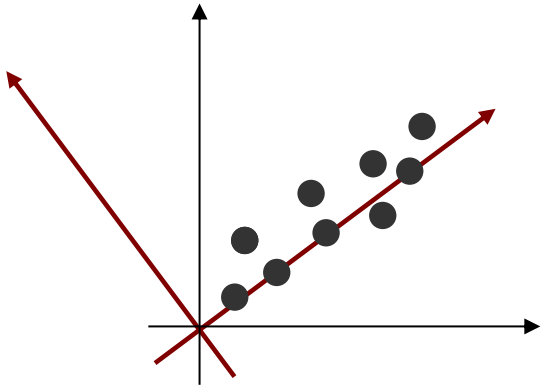
James Joseph Sylvester (1814-1897)
Camille Jordan (1838-1921)
Eugenio Beltrami (1835-1899)

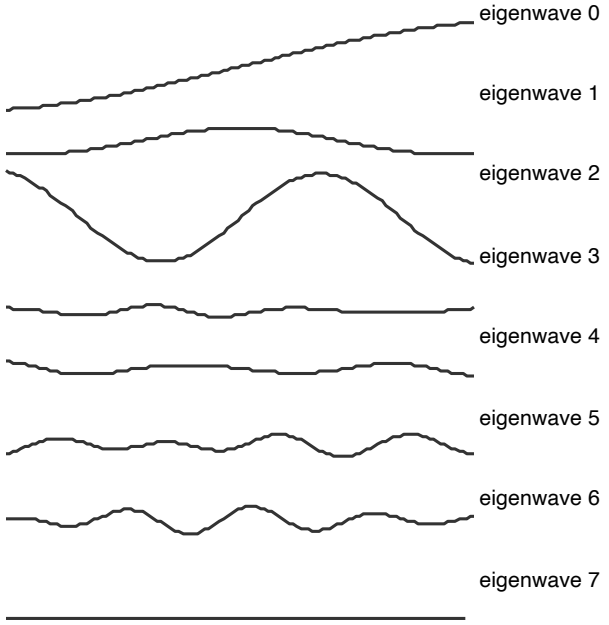
- Represent the time series as a linear combination of eigenwaves but keep only the first *N* coefficients.
- SVD is similar to Fourier and Wavelet approaches is that we represent the data in terms of a linear combination of shapes (in this case eigenwaves).
- SVD differs in that the *eigenwaves are data dependent*.





- We have previously seen that we can regard time series as points in high dimensional space.
- We can rotate the axes such that axis 1 is aligned with the direction of maximum variance, axis 2 is aligned with the direction of maximum variance orthogonal to axis 1 etc.
- Since the first few eigenwaves contain most of the variance of the signal, the rest can be truncated with little loss.



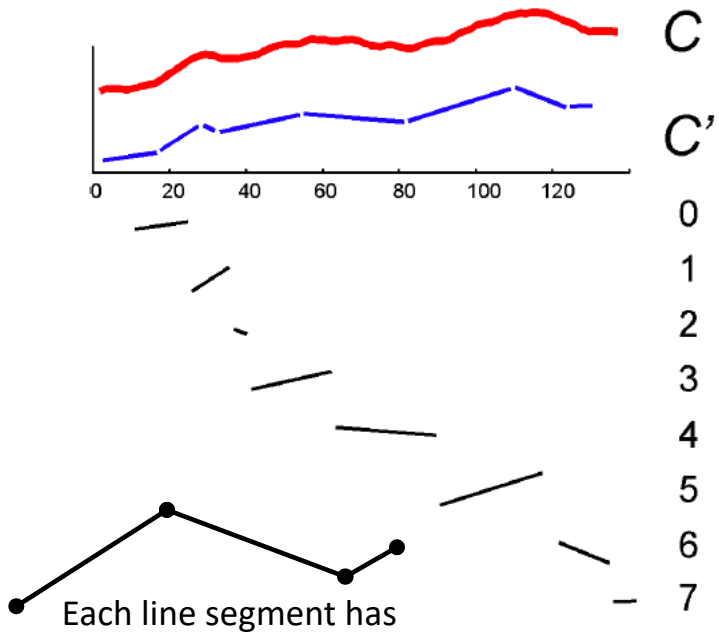


Piecewise Linear Approximation (PLA)

Karl Friedrich Gauss

1777 - 1855

- Represent the time series as a sequence of straight lines.
- Lines could be connected, in which case we are allowed K/2 lines, If lines are disconnected, we are allowed only K/3 lines
- In the literature there are numerous algorithms available for **segmenting** time series.
- An open question is how to best choose K, the "optimal" number of segments used to represent a particular time series.
- This problem involves a tradeoff between accuracy and compactness, and clearly has no general solution.
- Pros:
  - data compression
  - noise filtering
  - able to support some interesting non-Euclidean similarity measures



Each line segment has
- `length`
- `left_height`

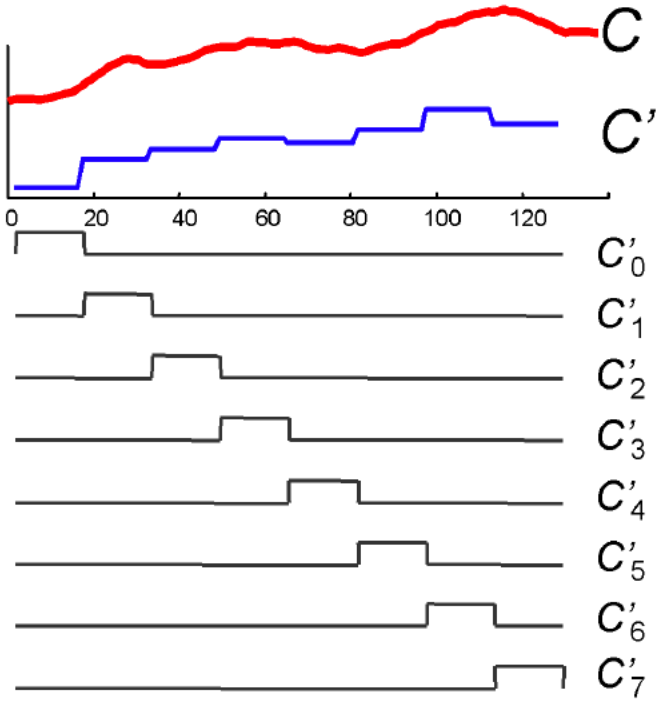
(`right_height` can be inferred by looking at the next segment)

Not available in Python

Piecewise Aggregate Approximation (PAA)

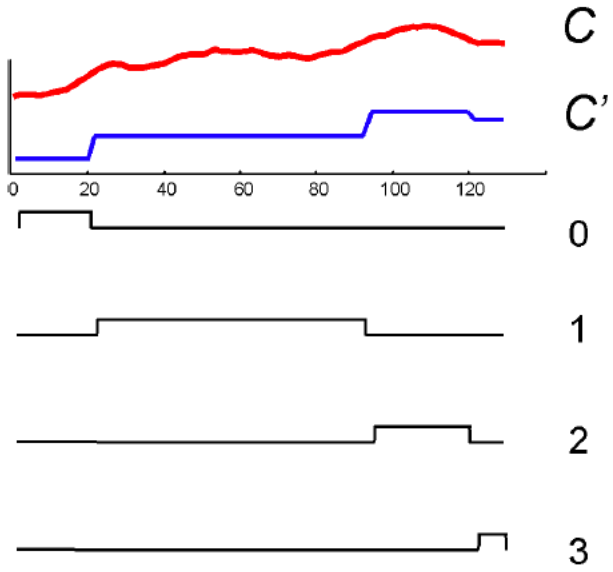
- Represent the time series as a sequence of box basis functions with each box of the same size.
- It approximates a TS by dividing it into equal-length segments and recording the mean value of the data points that fall within the segment.
- It reduces the data from *n* dimensions to *N* dimensions by dividing the time series into *N* equi-sized ``frames''.
- The mean value of the data falling within a frame is calculated, and a vector of these values becomes the data reduced representation.
- Pros
  - Extremely fast to calculate
  - Supports non Euclidean measures
  - Supports weighted Euclidean distance

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



Adaptive Piecewise Constant Approximation (APCA)

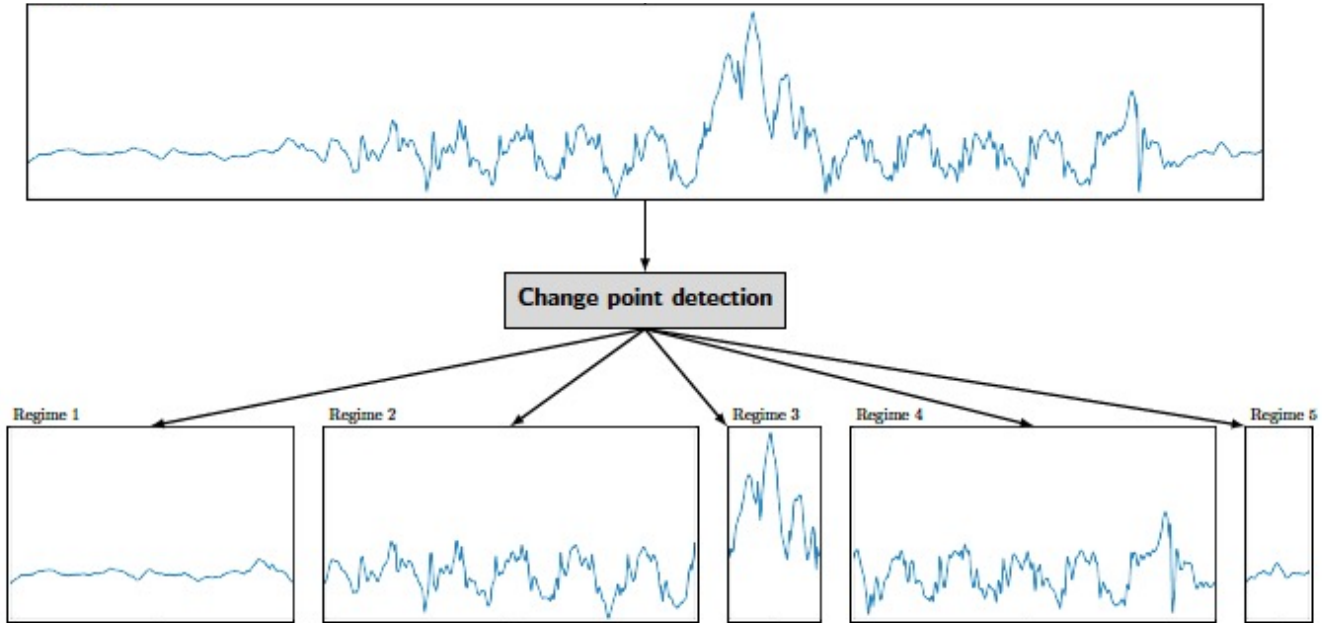
- It allows the segments to have arbitrary lengths, which in turn needs two numbers per segment.
- The first number records the mean value of all the data points in segment, and the second number records the length of the segment.
- APCA has the advantage of being able to place a single segment in an area of low activity and many segments in areas of high activity.
- In addition, one has to consider the structure of the data in question.
- Pros:
  - Fast to calculate O(n)
  - Supports non Euclidean measures
  - Supports weighted Euclidean distance



Not available in Python

Time Series Segmentation

- A TS can be segmented using *predefined length* ***w*** or *predefined number of segments* ***k,*** or by using **change point detection** methods.



- More details: Selective review of offline change point detection methods. Truong, C., Oudre, L., & Vayatis, N. (2020). Signal Processing, 167, 107299.

Symbolic Aggregate Approximation (SAX)

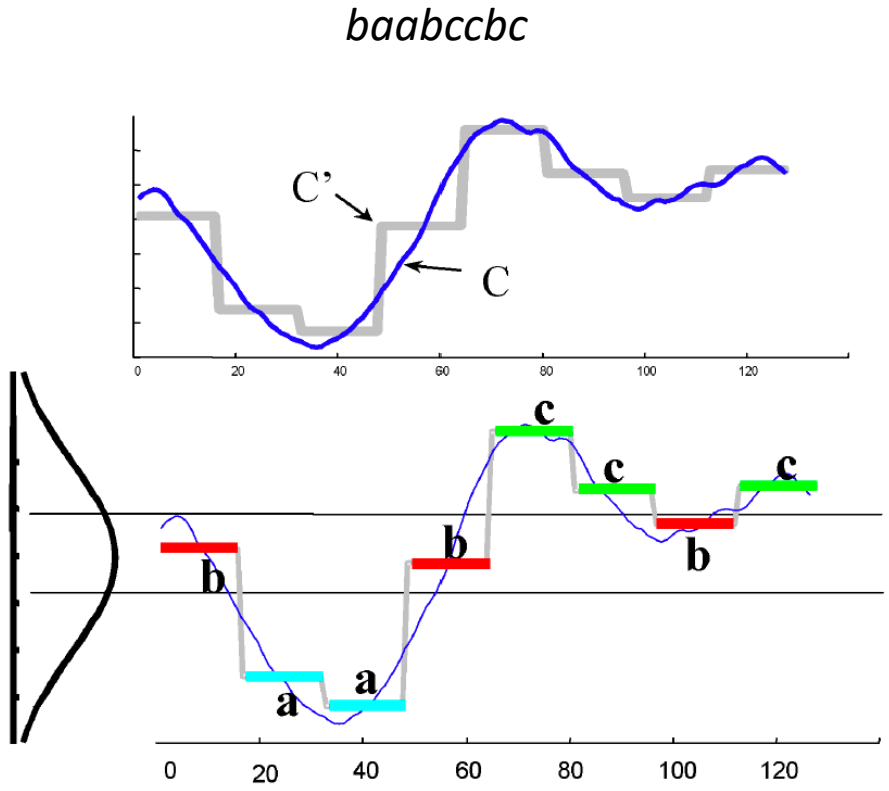
- Convert the data into a discrete format, with a *small alphabet size*.
- **Every part of the representation contributes about the same amount of information about the shape of the time series.**
- A time series ***T*** of length ***n*** is divided into ***w*** equal-sized segments; the values in each segment are then approximated and replaced by a single coefficient, which is their average. Aggregating these ***w*** coefficients form the PAA representation of **T**.

PAA

- Next, we determine the breakpoints that divide the distribution space into ***α*** equiprobable regions, where ***α*** is the alphabet size specified by the user (or it could be used the MDL).
- The breakpoints are determined such that the probability of a segment falling into any of the regions is approximately the same.
- If the symbols are not equi-probable, some of the substrings would be more probable than others. Consequently, we would inject a probabilistic bias in the process.

Symbolic Aggregate Approximation (SAX)

- Once the breakpoints are determined, each region is assigned a symbol.
- The PAA coefficients can then be easily mapped to the symbols corresponding to the regions in which they reside.
- The symbols are assigned in a bottom-up fashion, i.e., the PAA coefficient that falls in the lowest region is converted to “a”, in the one above to “b”, and so forth.



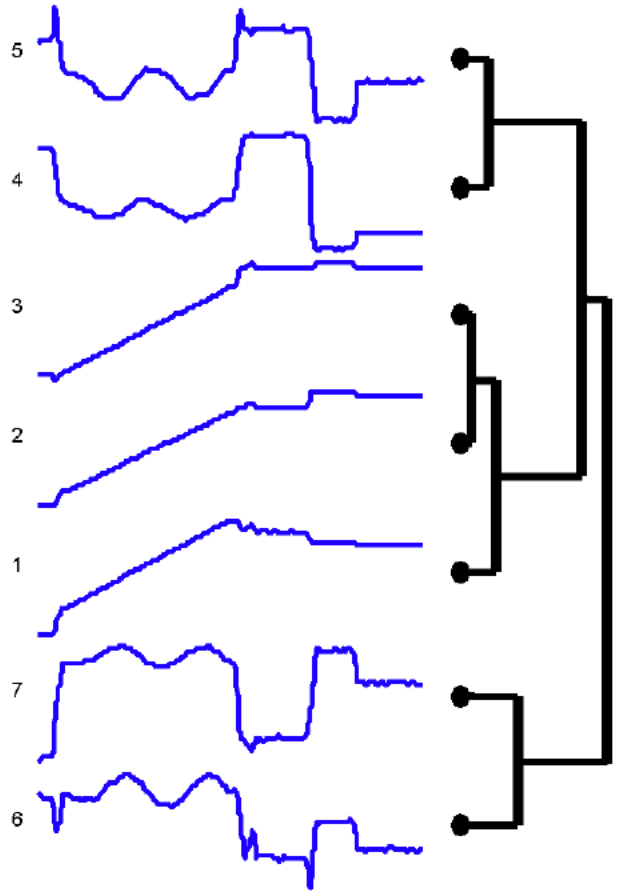
Clustering

Clustering Time Series

- It is based on the similarity between time series.
- The most similar data are grouped into clusters, but the clusters themselves should be dissimilar.
- These groups to find are not predefined, i.e., it is an unsupervised learning task.
- The two general methods of time series clustering are
  - Partitional Clustering and
  - Hierarchical Clustering

Hierarchical Clustering

- It ***computes pairwise distance***, and then merges similar clusters in a bottom-up fashion, without the need of providing the number of clusters
- It is one of the best tools to data evaluation, by creating a dendrogram of several time series from the domain of interest.
- Its application is limited to small datasets due to its quadratic computational complexity.



Partitional Clustering

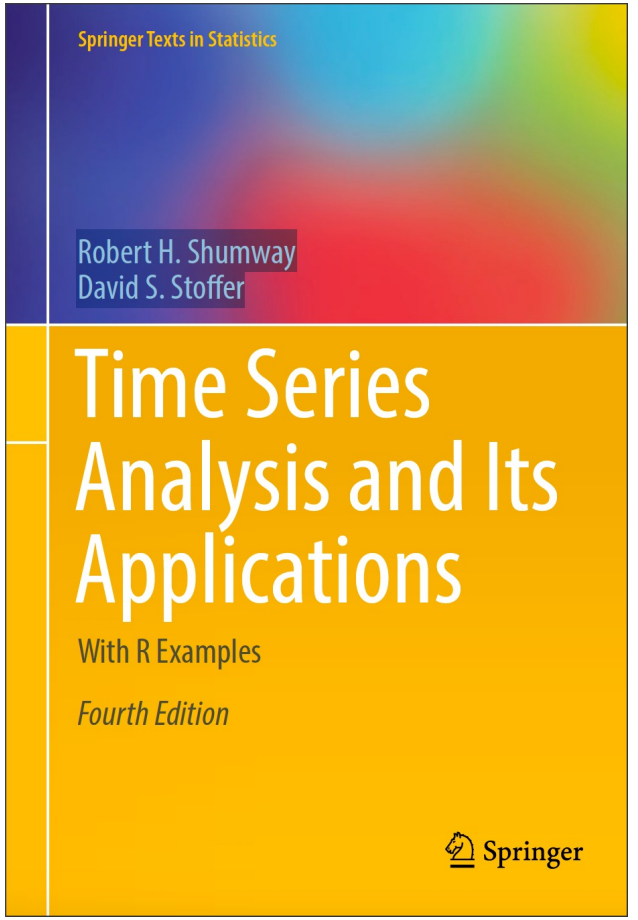
- Typically uses K-Means (or some variant) to optimize the objective function by minimizing the sum of squared intra-cluster errors.
- K-Means is perhaps the most commonly used clustering algorithm in the literature, one of its shortcomings is the fact that the number of clusters, K, must be pre-specified.
- Also, the ***distance function plays a fundamental role*** both for the quality of the results and for the efficiency.

Types of Time Series Clustering

- ***Whole clustering***: similar to that of conventional clustering of discrete objects. Given a set of individual time series data, the objective is to group similar time series into the same cluster.
- ***Features-based clustering***: extract features, or time series motifs (see next lectures) as the features and use them to cluster time series.
- ***Compression-based clustering***: compress time series and run clustering on the compressed versions.
- ***Subsequence clustering***: given a single time series, subsequence clustering is performed on each individual time series extracted from the long time series with a sliding window.

References

- Selective review of offline change point detection methods. Truong, C., Oudre, L., & Vayatis, N. (2020). Signal Processing, 167, 107299.
- Time Series Analysis and Its Applications. Robert H. Shumway and David S. Stoffer. 4th edition.(https://www.stat.pitt.edu/stoffer/tsa4/tsa4.pdf)
- Mining Time Series Data. Chotirat Ann Ratanamahatana et al. 2010. (https://www.researchgate.net/publication/227001229_Mining_Time_Series_Data)
- Dynamic Programming Algorithm Optimization for Spoken Word Recognition. Hiroaki Sakode et al. 1978.
- Experiencing SAX: a Novel Symbolic Representation of Time Series. Jessica Line et al. 2009
- Compression-based data mining of sequential data. Eamonn Keogh et al. 2007.



Exercises Approximation

DWT – Exercise 1

- Given the following input time series: $56, \quad 40, \quad 8, \quad 24, \quad 48, \quad 48, \quad 40, \quad 16$
- Compute the DWT.
- Approach: First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the means. Differences are kept in each step.

direct transform $(a, b) \rightarrow (d, s)$

$$s = \frac{a+b}{2},$$

$$d = a - s.$$

inverse $(d, s) \rightarrow (a, b)$

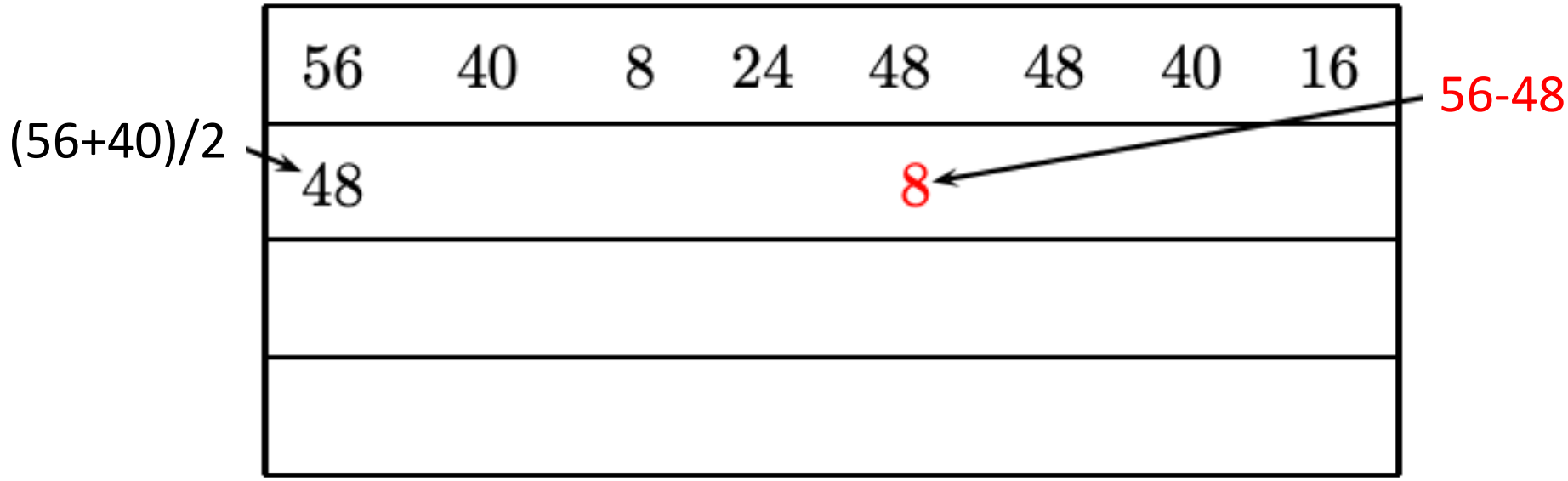
$$a = s + d;,$$

$$b = s - d.$$

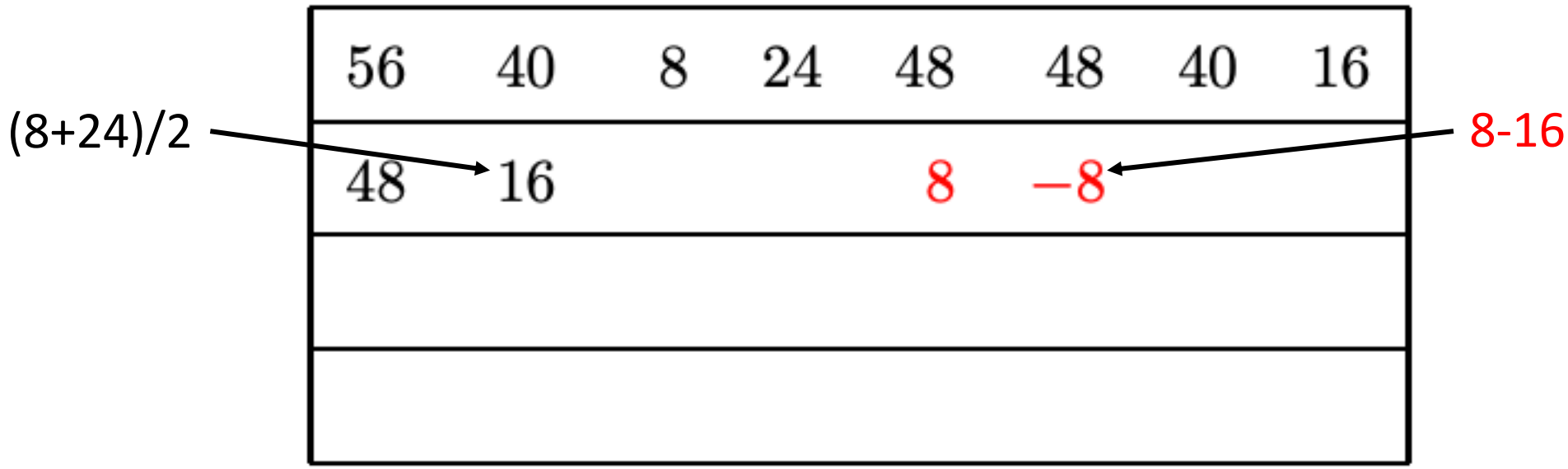
DWT – Exercise 1 - Solution

56	40	8	24	48	48	40	16

DWT – Exercise 1 - Solution



DWT – Exercise 1 - Solution



DWT – Exercise 1 - Solution

56	40	8	24	48	48	40	16
48	16	48		8	−8	0	

DWT – Exercise 1 - Solution

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12

DWT – Exercise 1 - Solution

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
				8	−8	0	12

DWT – Exercise 1 - Solution

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32		16		8	−8	0	12

DWT – Exercise 1 - Solution

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	8	−8	0	12

DWT – Exercise 1 - Solution

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	8	−8	0	12
		16	10	8	−8	0	12

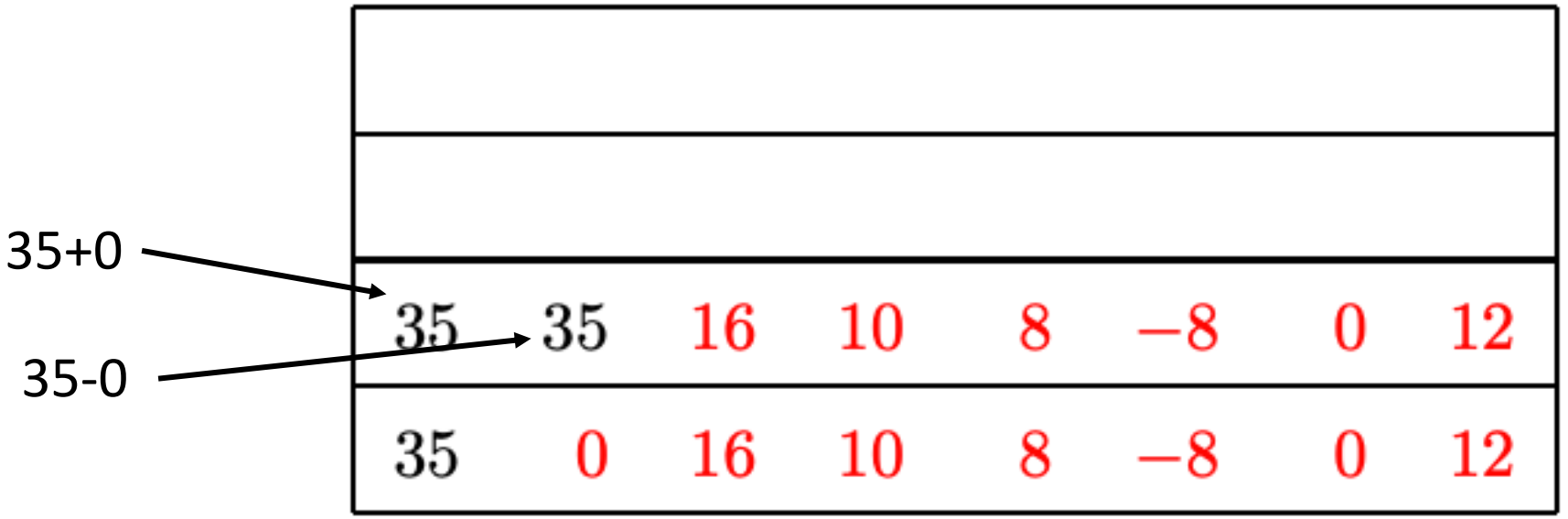
DWT – Exercise 1 - Solution

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	8	−8	0	12
35	−3	16	10	8	−8	0	12

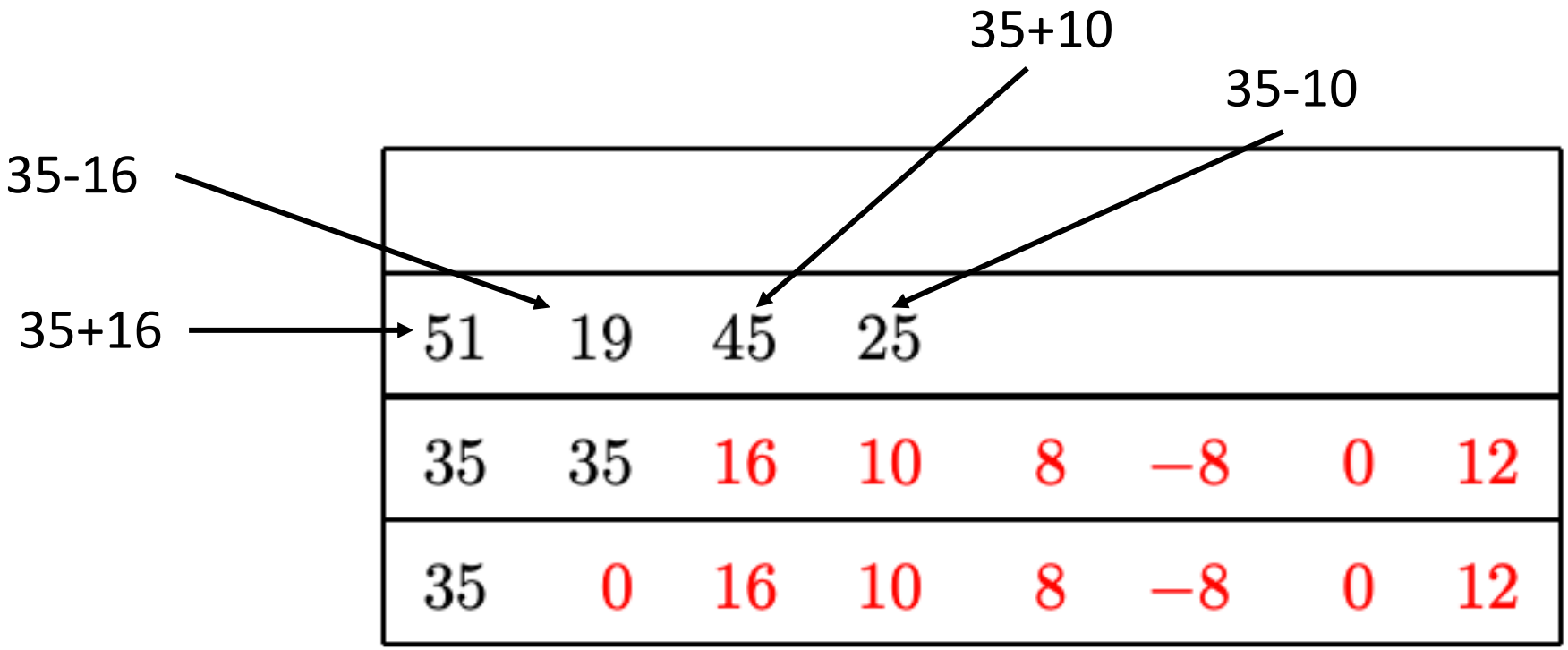
DWT – Exercise 2

- The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean, and repeat the process up to the first row.

DWT – Exercise 2 Solution



DWT – Exercise 2 Solution



DWT – Exercise 2 Solution

59	43	11	27	45	45	37	13
51	19	45	25	8	−8	0	12
35	35	16	10	8	−8	0	12
35	0	16	10	8	−8	0	12