电子科技大学
University of Electronic Science and Technology of China

Symbolic Aggregate ApproXimation

Heng Zhang

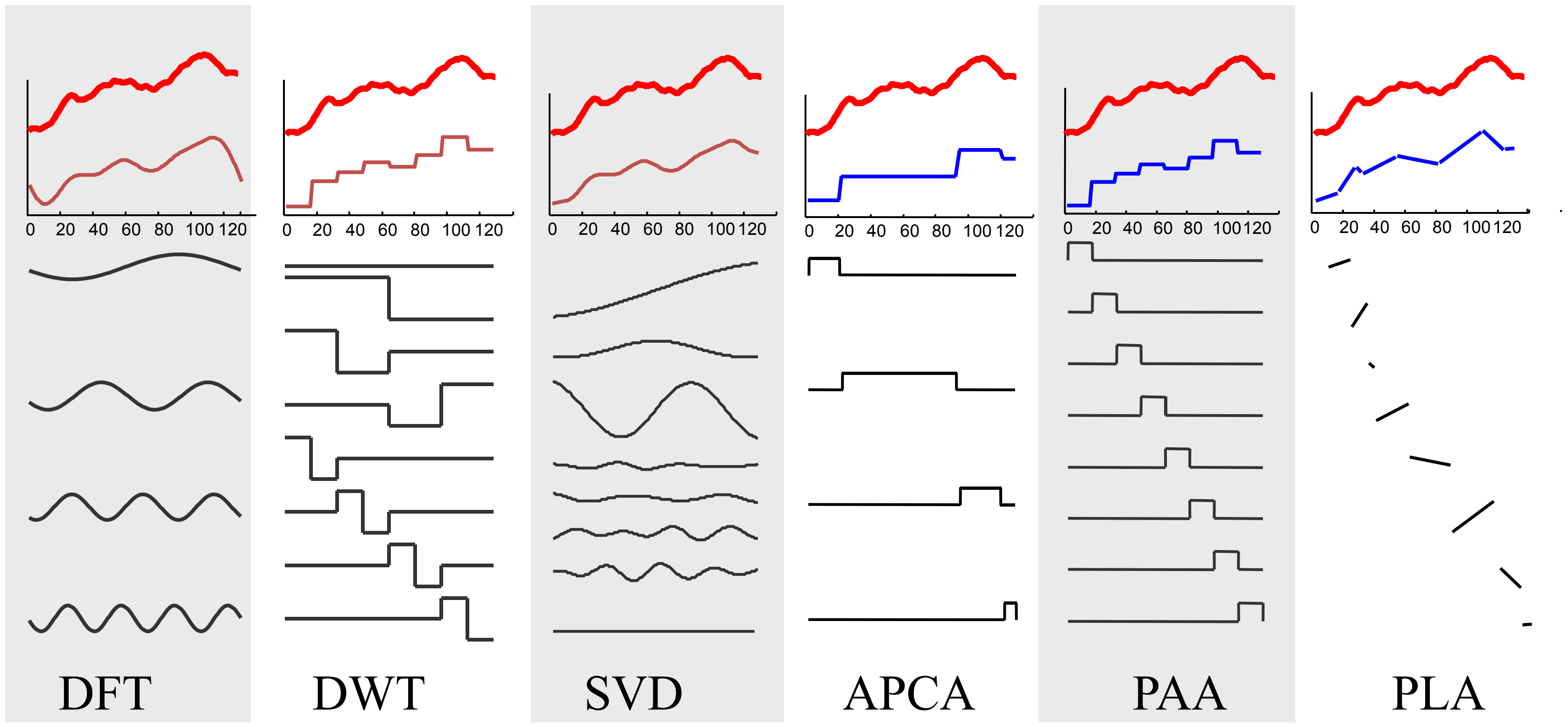
Data Mining Lab, Big Data Research Center, UESTC
Email： junmshao@uestc.edu.cn
http://staff.uestc.edu.cn/shaojunming

Outline

数据挖掘实验室
Data Mining Lab

1. The Common Representations of Time Series
2. **S**ymbolic **A**ggregate Appro**X**imation
3. Lower Bounding Distance Measure
4. Using SAX to Find Discords and Motifs of Time Series

1. Common Representations of Time Series



1.1Discrete Fourier Transform

Consider the following expression vector:

$$\begin{bmatrix} 23 \\ -11 \\ 6 \end{bmatrix} = 6 * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + (-11) * \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 23 * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The Orthogonal basis is unique?

$$\begin{bmatrix} 23 \\ -11 \\ 6 \end{bmatrix} = 7 * \begin{bmatrix} \frac{2}{7} \\ \frac{3}{7} \\ \frac{6}{7} \end{bmatrix} + 14 * \begin{bmatrix} \frac{6}{7} \\ \frac{2}{7} \\ -\frac{3}{7} \end{bmatrix} + 21 * \begin{bmatrix} \frac{3}{7} \\ -\frac{6}{7} \\ \frac{2}{7} \end{bmatrix} \quad \ldots$$

1.1Discrete Fourier Transform

So we can use the following equation:

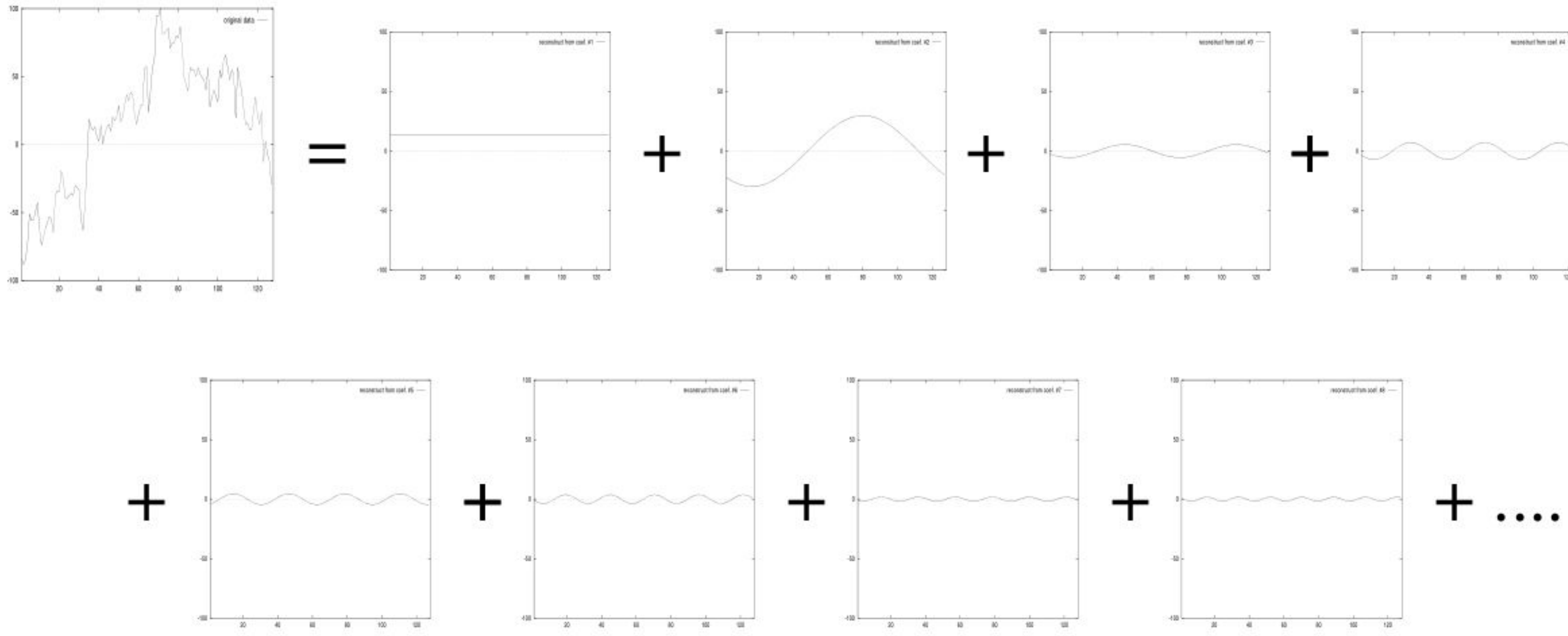
$$v(t) = a_1 v_1(t) + a_2 v_2\,(t) + \ldots + a_k v_k(t)$$

$$\Big\downarrow \; f, v \in R^{2k}$$

$$f(t) = \frac{1}{2} a_0 + a_1 v_1(t) + a_2 v_2(t) + \cdots + a_k v_k(t) + b_1 w_1(t) + b_2 w_2(t) + \cdots + b_k w_k(t)$$

$$\Big\downarrow$$

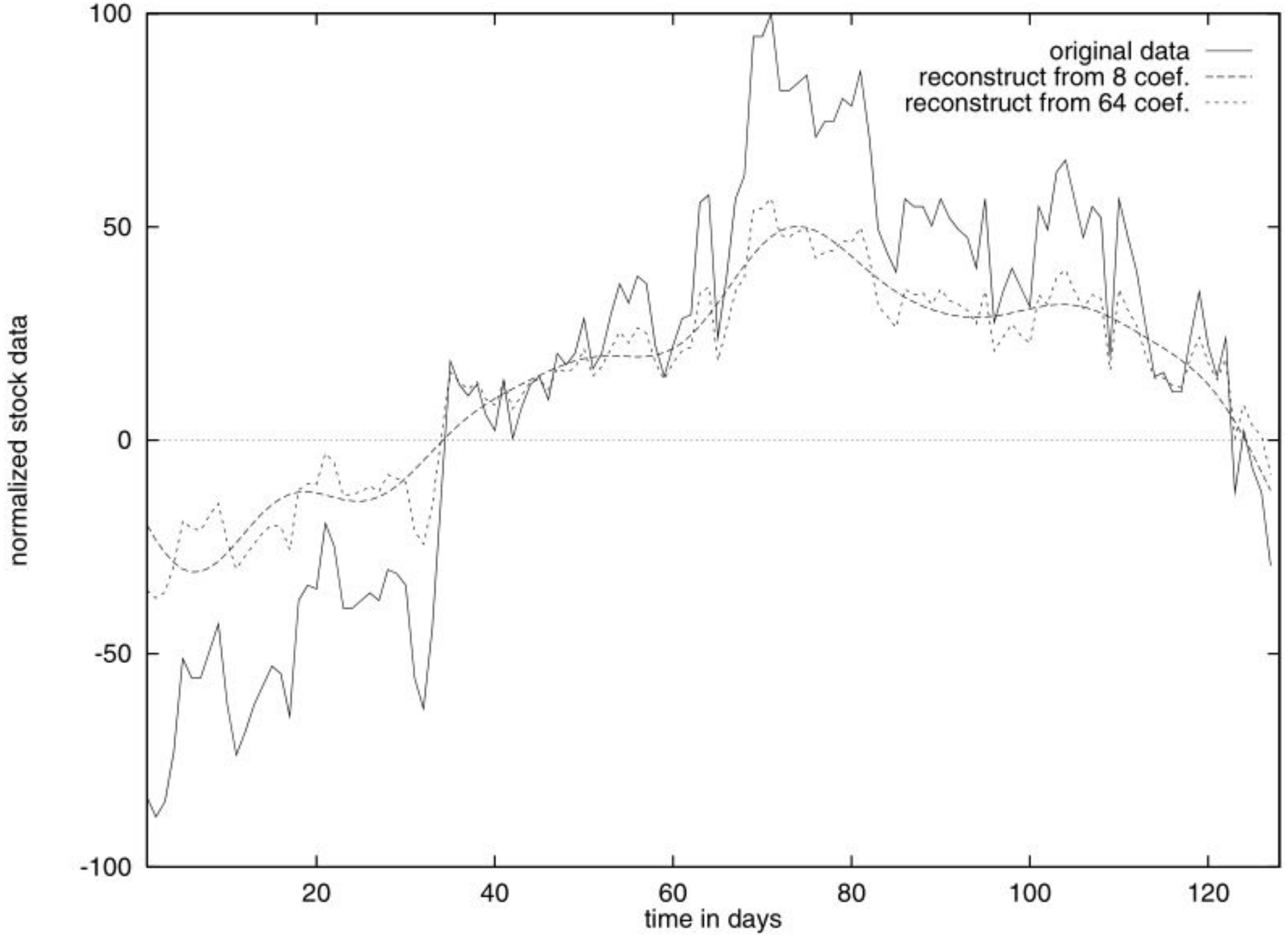
$$f(t) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} a_n \cos(nt) + \sum_{n=1}^{\infty} b_n \sin(nt)$$



Discrete Fourier Transform

数据挖掘实验室

Data Mining Lab

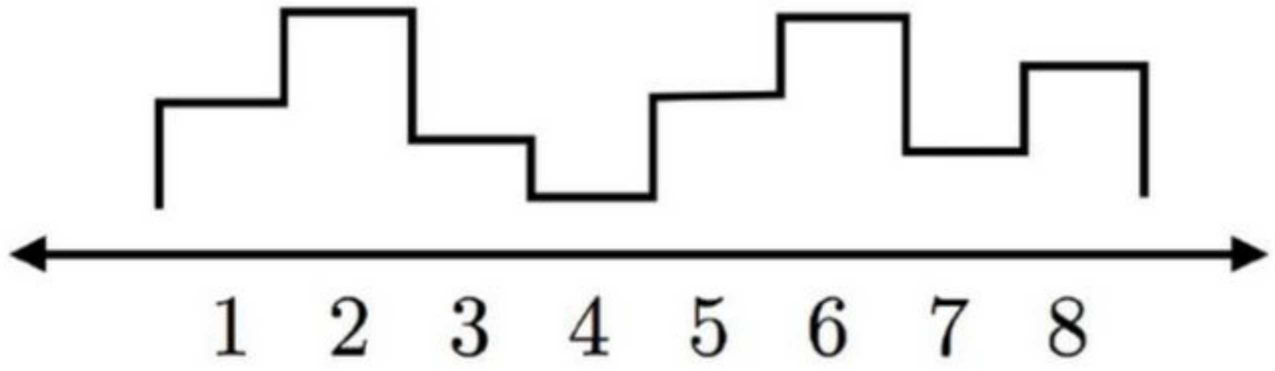


1.2 Discrete Wavelet Transform

Discrete Wavelet Transform performs similar properties as DFT in time series . Whereas the basis function of DFT is sinusoid the wavelet function are defined by wavelet basis.

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \varphi(t-k) + \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} d_{j,k} \Psi(2^j t - k)$$

1.2 Discrete Wavelet Transform



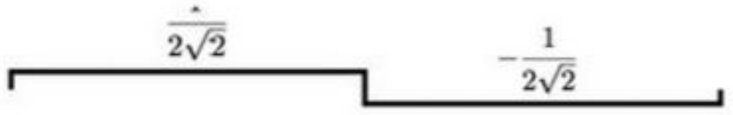
A time series, length = 8

1.2 Discrete Wavelet Transform

$$\psi(n) = \begin{cases} \frac{1}{2\sqrt{2}} & 1 \le n \le 4 \\ -\frac{1}{2\sqrt{2}} & 5 \le n \le 8 \\ 0 & \text{otherwise} \end{cases}$$

Basic wavelet function, it is **orthogonal** with itself

$\psi(n)$:



1.2 Discrete Wavelet Transform

In order to translation and scaling, transform it to $\Psi(2n)$

$$\psi(2n) = \begin{cases} \frac{1}{2\sqrt{2}} & 1 \leq n \leq 2 \\ -\frac{1}{2\sqrt{2}} & 3 \leq n \leq 4 \\ 0 & \text{otherwise} \end{cases}$$

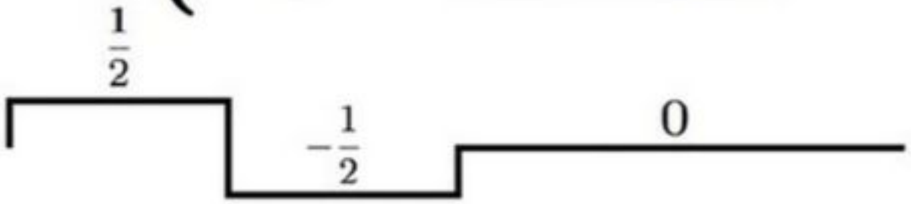
However, it doesn’t meet the inner product = 1

1.2 Discrete Wavelet Transform

数据挖掘实验室
Data Mining Lab

$$\sqrt{2}\psi(2n) = \begin{cases} \frac{1}{2} & 1 \le n \le 2 \\ -\frac{1}{2} & 3 \le n \le 4 \\ 0 & \text{otherwise} \end{cases}$$

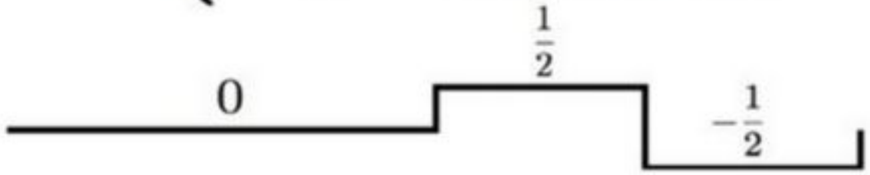
$\sqrt{2}\psi(2n)$:



The length is 8 , so translation it:

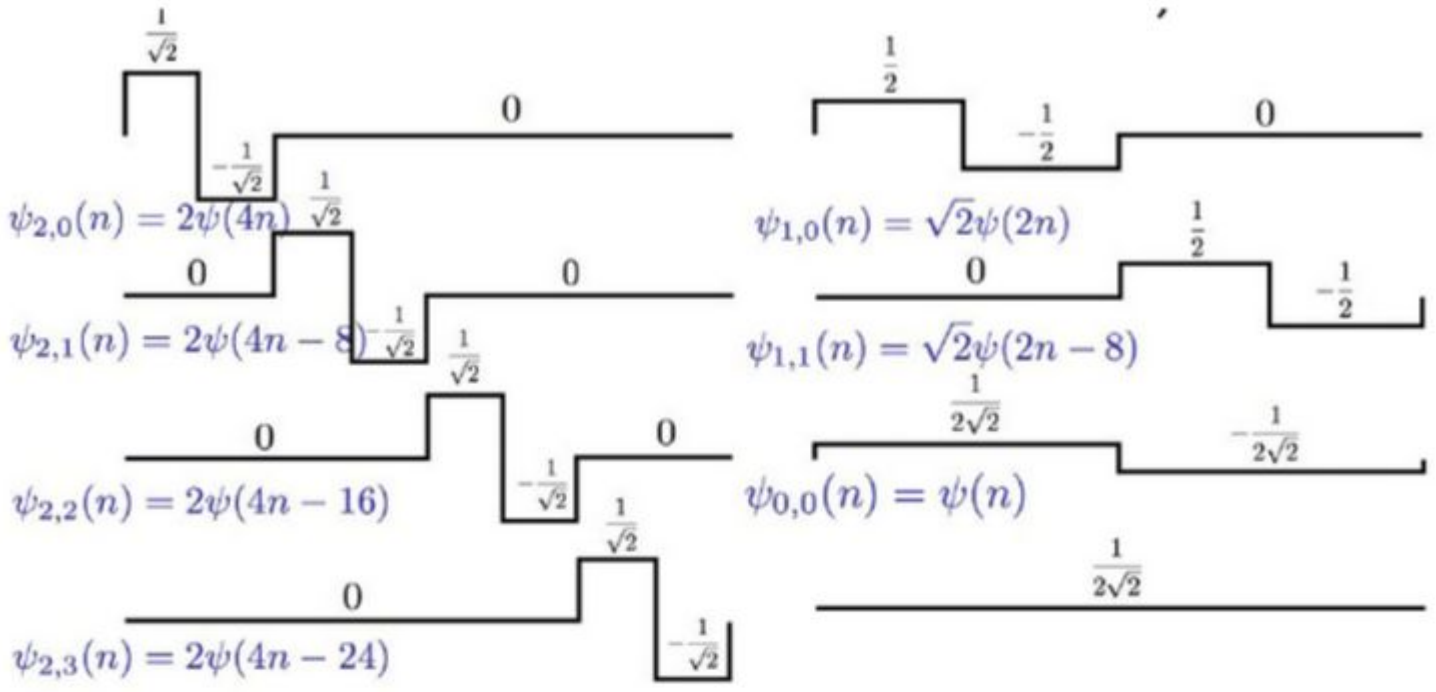
$$\sqrt{2}\psi(2n-8) = \begin{cases} \frac{1}{2} & 5 \le n \le 6 \\ -\frac{1}{2} & 7 \le n \le 8 \\ 0 & \text{otherwise} \end{cases}$$

$\sqrt{2}\psi(2n-8)$:

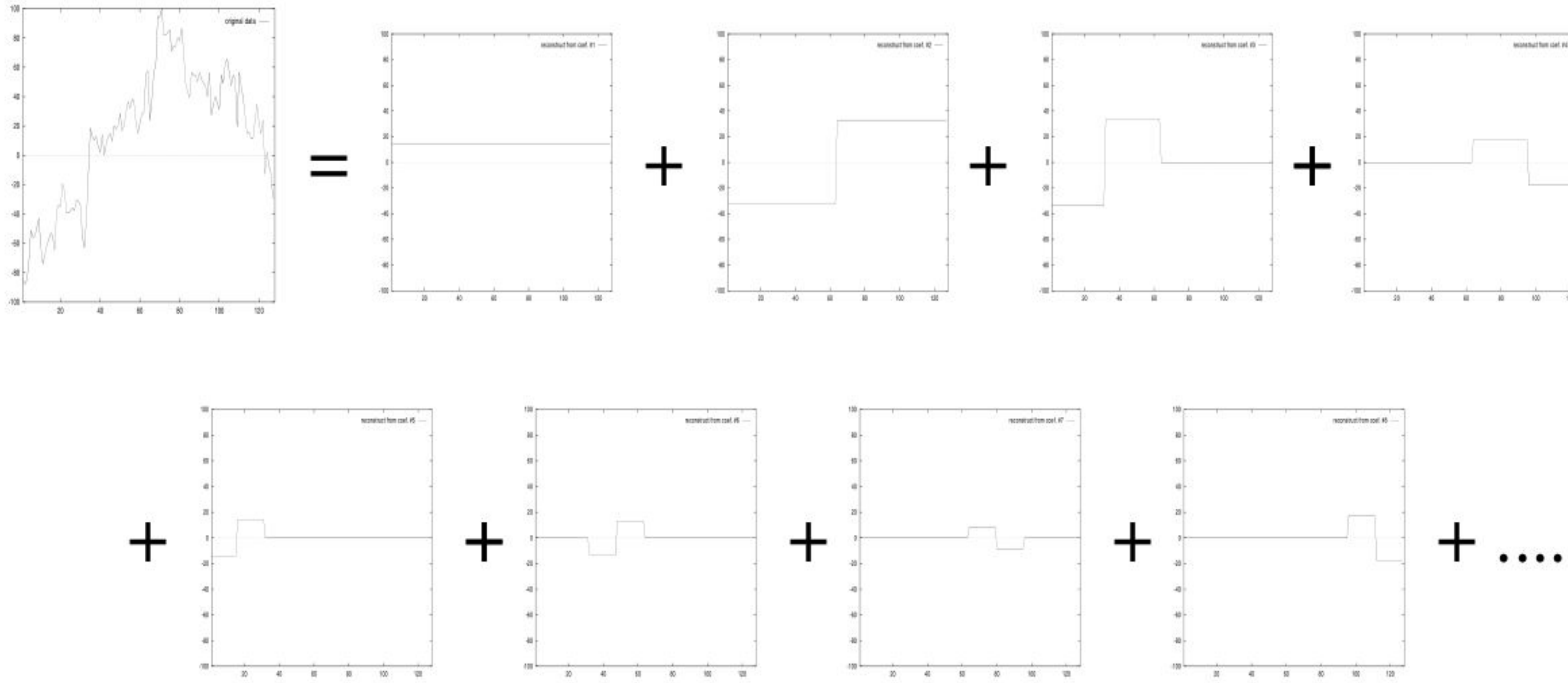


1.2 Discrete Wavelet Transform

Finally the result as follow:

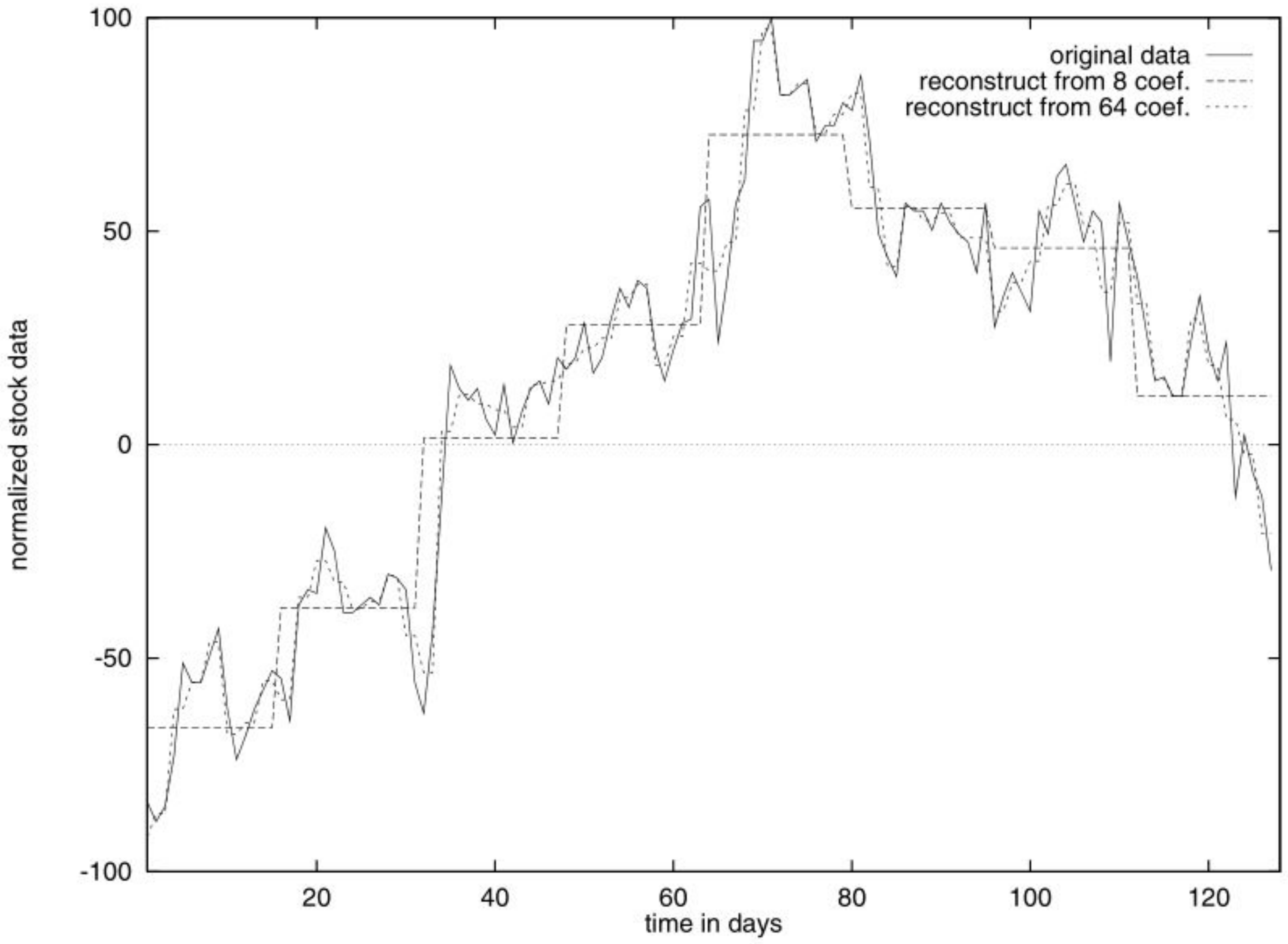


$\frac{1}{\sqrt{2}}$
$-\frac{1}{\sqrt{2}}$
0
$\psi_{2,0}(n) = 2\psi(4n)$
$\psi_{2,1}(n) = 2\psi(4n-8)$
$\psi_{2,2}(n) = 2\psi(4n-16)$
$\psi_{2,3}(n) = 2\psi(4n-24)$
$\frac{1}{2}$
$-\frac{1}{2}$
$\psi_{1,0}(n) = \sqrt{2}\psi(2n)$
$\psi_{1,1}(n) = \sqrt{2}\psi(2n-8)$
$\frac{1}{2\sqrt{2}}$
$-\frac{1}{2\sqrt{2}}$
$\psi_{0,0}(n) = \psi(n)$

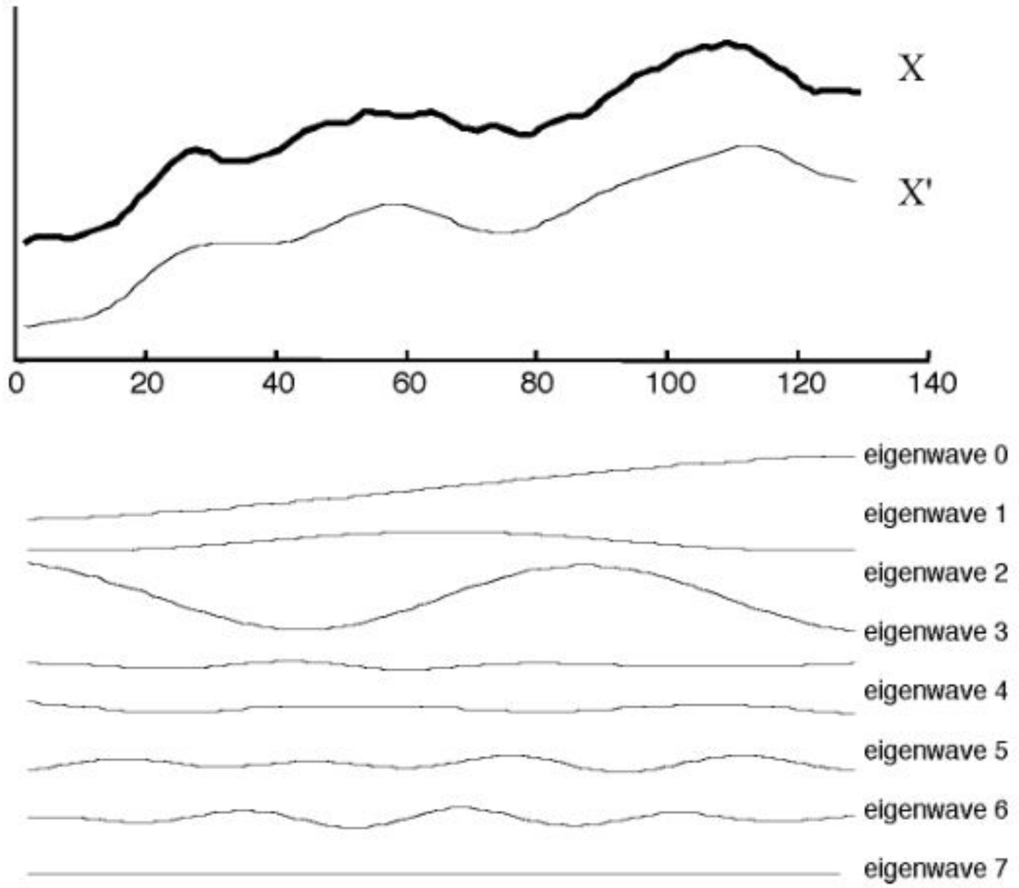


Discrete Wavelet Transform

数据挖掘实验室
Data Mining Lab



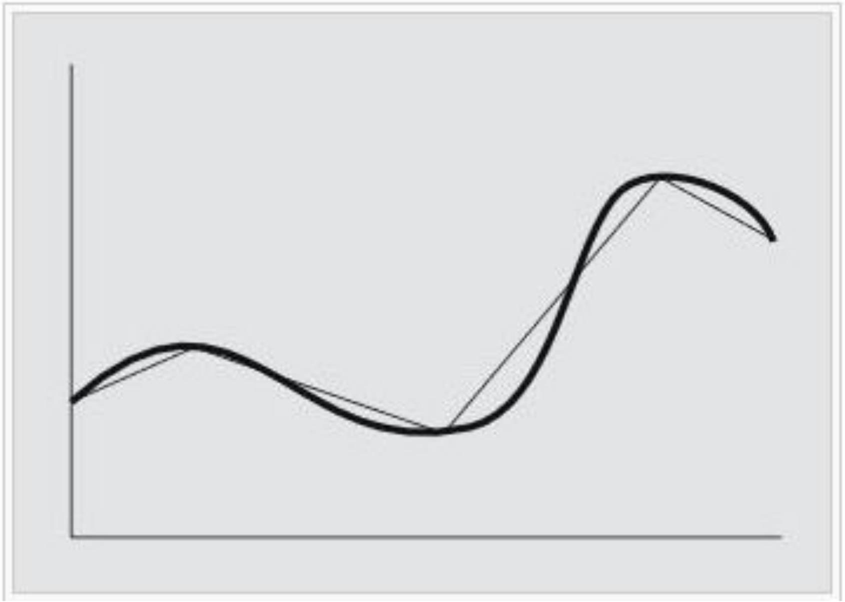
SVD



$$M = U\Sigma V^T$$

PLA

Piecewise Linear Approximation



$$f(x) = \begin{cases} -0.477x + 1 & x \in \ [0, \pi/3] \\ -0.955x + 0.5 & x \in \ [\pi/3, 2\pi/3] \\ -0.477x + 1.5 & x \in \ [2\pi/3, 1] \end{cases} .$$

Disadvantage of DFT and DWT:

Only focus on the fidelity of approximation.

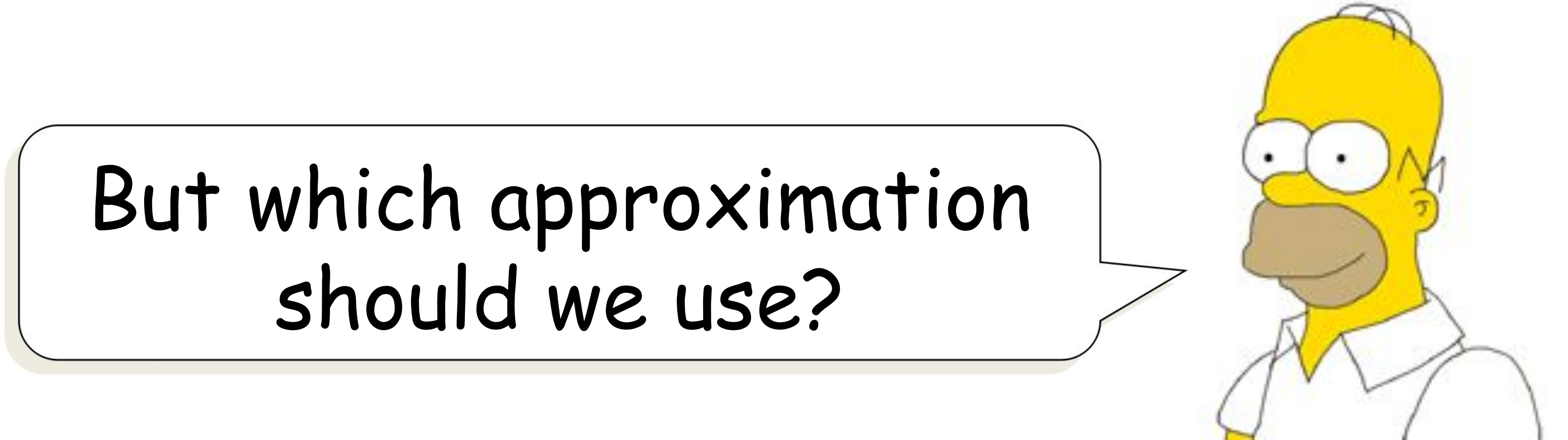
It’s difficult to measure the similarity between two different length of time series.

The Generic Data Mining Algorithm of time series

• Create an approximation of the data, which will fit in main memory

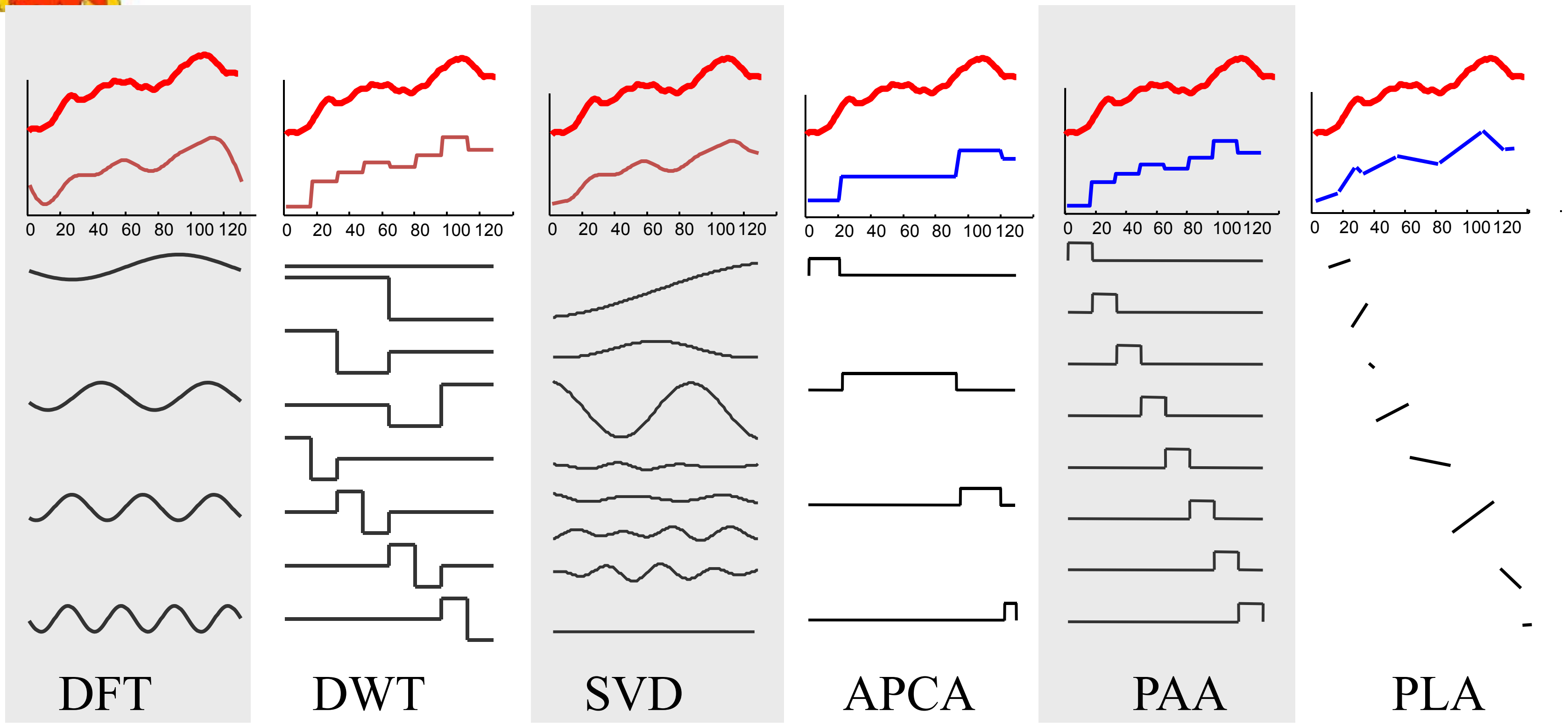
• Approximately solve the problem at hand in main memory

• Make accesses to the original data on disk to confirm the solution



Some approximations of time series...

..note that all are real valued...

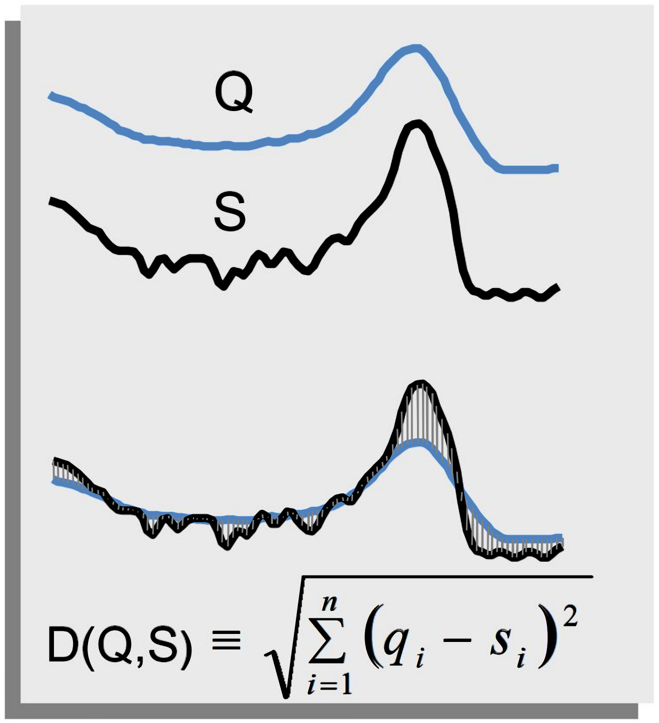


The Generic Data Mining Algorithm of time series

This only works if the approximation allows **lower bounding**

What is Lower Bounding?

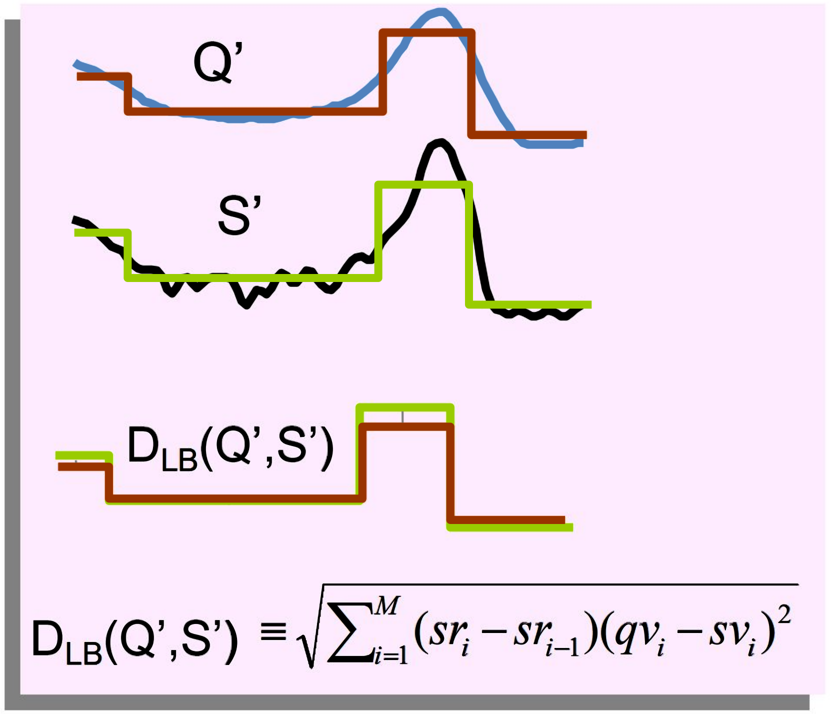
• Lower bounding means the estimated distance in the reduced space is always less than or equal to the distance in the original space.



$$D(Q,S) \equiv \sqrt{\sum_{i=1}^{n}\left(q_i - s_i\right)^2}$$

Raw Data

Approximation or "Representation"



$$D_{LB}(Q',S') \equiv \sqrt{\sum_{i=1}^{M}(sr_i - sr_{i-1})(qv_i - sv_i)^2}$$

Lower bounding means that for all Q and S, we have: $D_{LB}(Q',S') \leq D(Q,S)$

Lower Bounding functions are known for wavelets, Fourier, SVD, piecewise polynomials, Chebyshev Polynomials and clipped data

While there are more than 200 different symbolic or discrete ways to approximate time series, none except SAX allows lower bounding

Why do we care so much about symbolic representations?

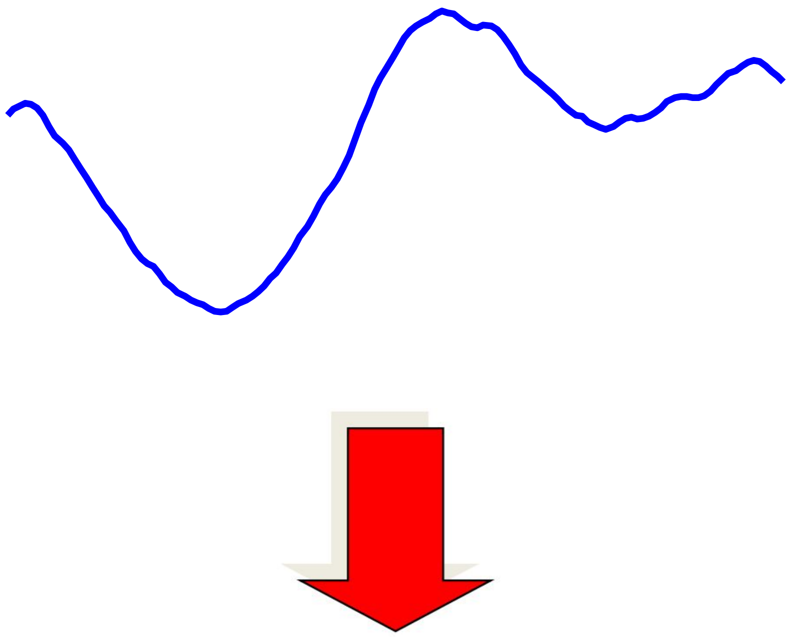
Symbolic Representations Allow:

- Hashing
- Suffix Trees
- Markov Models
- Stealing ideas from text processing/ bioinformatics community
- etc

There is *one* symbolic representation of time series, that allows…

- Lower bounding of Euclidean distance
- Lower bounding of the DTW distance
- Dimensionality Reduction
- Numerosity Reduction

2.Symbolic Aggregate ApproXimation



baabccbc

2.1 Dimensionality Reduction via PAA

数据挖掘实验室

Data Mining Lab

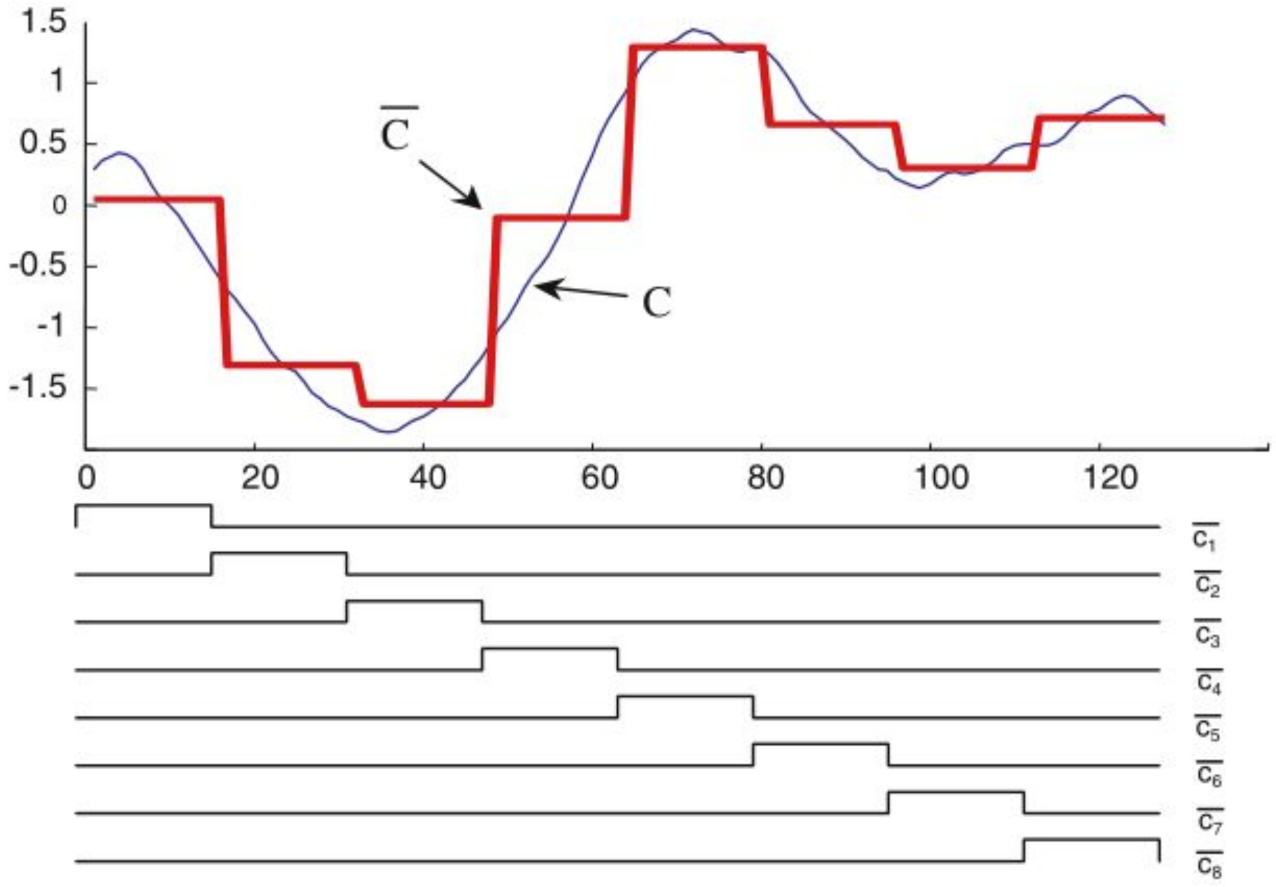
A time series C of length n can be represented in a w- dimensional space by a vector $\bar{C} = \bar{c}_1, \ldots, \bar{c}_w$. The i -th element of $\bar{C}$ is calculated by the following equation:

$$\bar{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j$$

To reduce the time series from n dimension to w dimensions, the data is divided into w equal sized frames. The mean of the data falling within a frame is calculated and a vector of these values becomes the data-reduced representation.

2.1 Dimensionality Reduction via PAA

数据挖掘实验室
Data Mining Lab



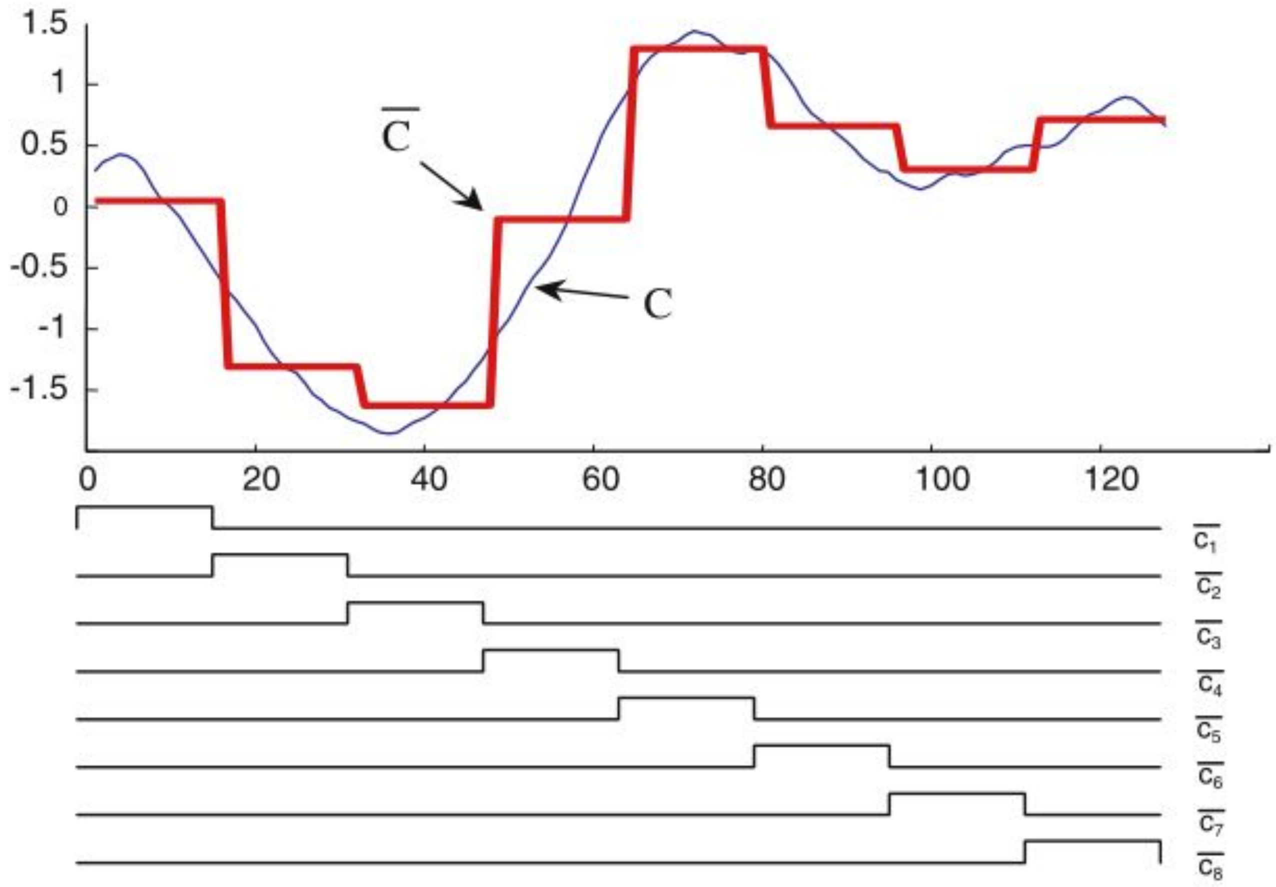
The PAA representation can be visualized as an attempt to model a time series with a linear combination of box basis functions . In this case, a sequence of length 128 is reduced to 8 dimensions.

2.2 Discretization

Having transformed a time series database into the PAA we can apply a further transformation to obtain a discrete representation.

It is desirable to have a discretization technique that will produce symbols with **equiprobability**.

This is easily achieved since normalized time series have a **Gaussian** distribution.



In the y-axis the value of time series have a **Gaussian** distribution, when the number are growthing.

2.2 Discretization

数据挖掘实验室 Data Mining Lab

Breakpoints: breakpoints are a sorted list of numbers B= $\beta_1,\ldots,\beta_{a-1}$ such that the area under a $N(0,1)$ Gaussian curve from β_i to $\beta_{i+1} = 1/a$ (β_0 and β_a are defined as $-\infty$ and ∞, respectively).

β_i \ a	3	4	5	6	7	8	9	10
β_1	−0.43	−0.67	−0.84	−0.97	−1.07	−1.15	−1.22	−1.28
β_2	0.43	0	−0.25	−0.43	−0.57	−0.67	−0.76	−0.84
β_3		0.67	0.25	0	−0.18	−0.32	−0.43	−0.52
β_4			0.84	0.43	0.18	0	−0.14	−0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

A look up table that contains the breakpoints that divide a Gaussian distribution in an arbitrary number(from 3 to 10) of equiprobability regions .

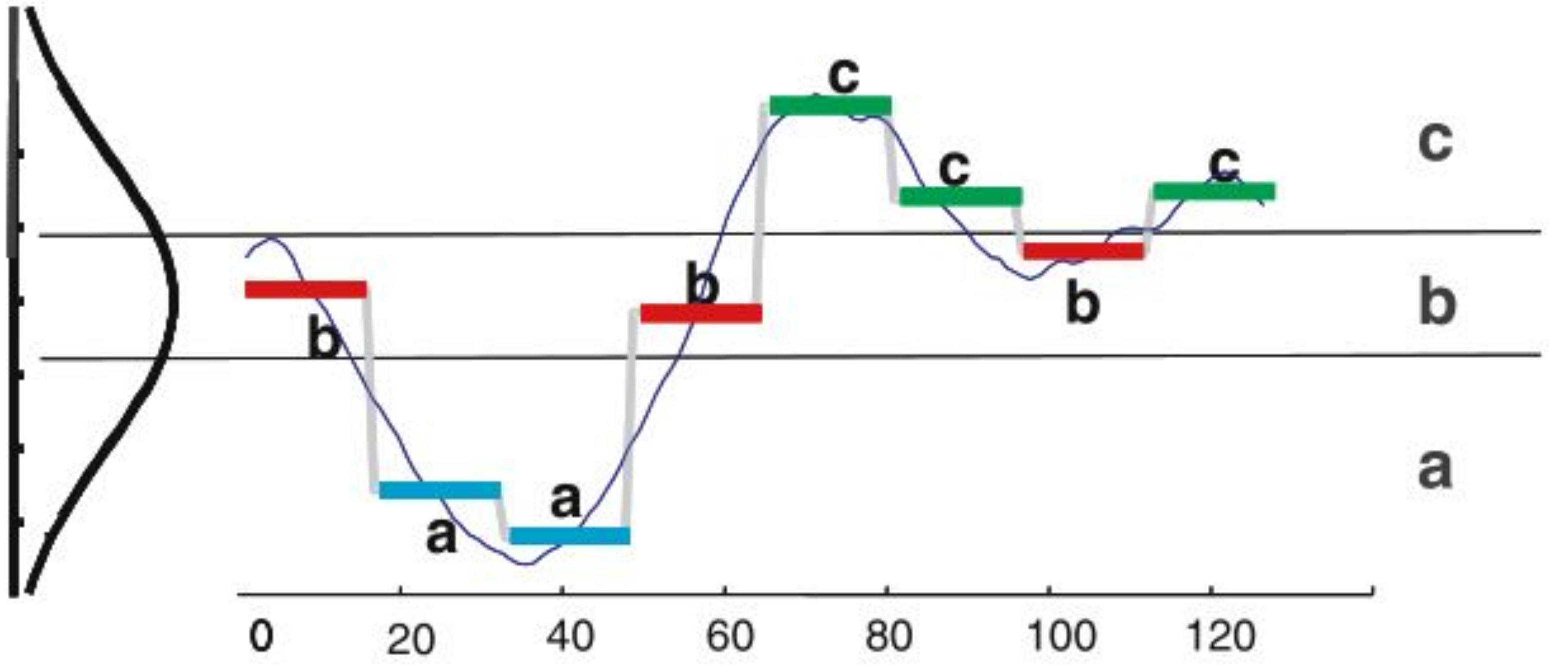
2.2 Discretization

数据挖掘实验室
Data Mining Lab

Words: A subsequence C of length n can be represented as a word $C = \hat{c}_1, \ldots, \hat{c}_w$ as follows. Let alpha i denote the *i-th* element of the alphabet, i.e.,$alpha_1 = a$ and $alpha_2 = b$. Then the mapping from a PAA approximation $\bar{C}$ to a word $\hat{C}$ is obtained as follows:

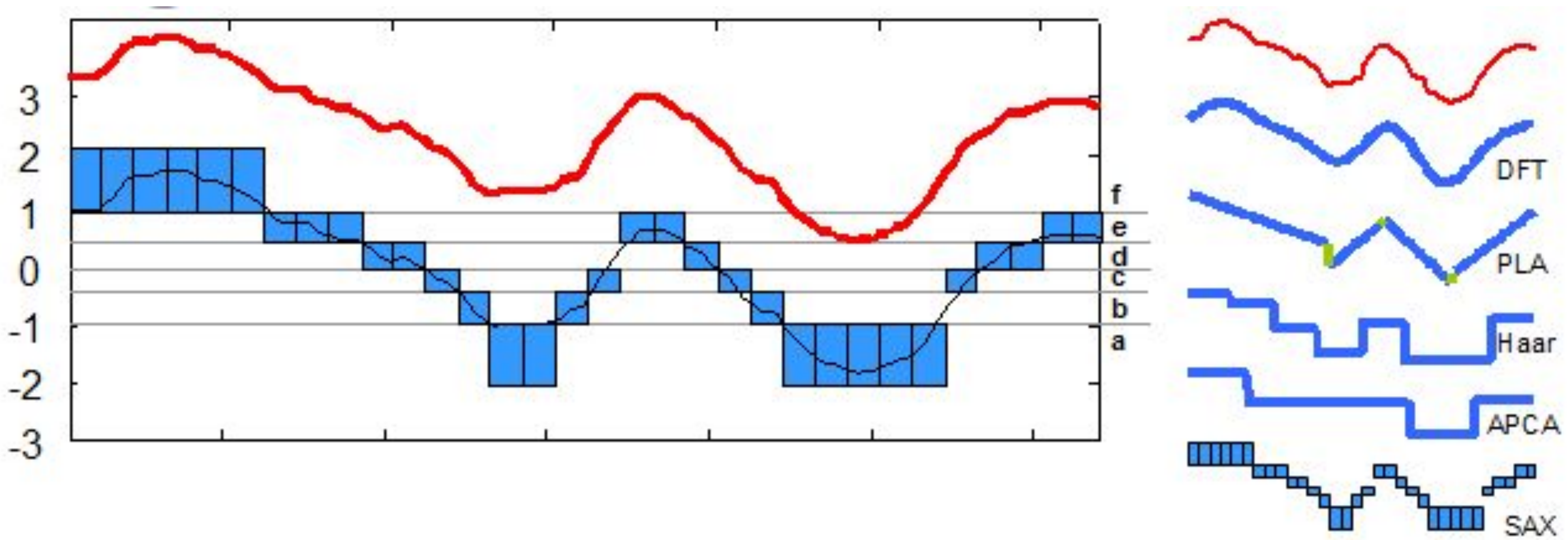
$$\hat{c}_i = alpha_j \quad , \qquad iif \quad \beta_{j-1} \leq \hat{c}_i \leq \beta_j$$

2.2 Discretization



A time series is discretized by first obtaining a PAA approximation and then using predetermined breakpoints to map the PAA coefficients into SAX symbols. In the example above, with n = 128, w = 8, and a =3 , the time series is mapped to the word **baabccbc**.

Visual Comparison



A raw time series of length 128 is transformed into the word "**ffffffeeeddcbaabceedcbaaaaacddee**."

– We can use more symbols to represent the time series since each symbol requires fewer bits than real-numbers (float, double)

3. Distance Measures

Given two time series Q and C of the same length n, the following Equation defines their Euclidean distance:

$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2}$$

If we transform the original subsequences into PAA representations, $\overline{Q}$ and $\overline{C}$, similar with the above Eq, we can then obtain a lower bounding approximation of the Euclidean distance between the original subsequences by:

$$DR(\overline{Q},\overline{C}) \equiv \sqrt{\frac{n}{w}} \sqrt{\sum\nolimits_{i=1}^{w} (\overline{q}_i - \overline{c}_i)^2}$$

3. Distance Measures

数据挖掘实验室
Data Mining Lab

If we further transform the data into the symbolic representation, we can define a MINDIST function that returns the minimum distance between the original time series of two words:

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

The *dist()* function can be implemented using a lookup table:

	a	**b**	**c**	**d**
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

This table is for an alphabet of cardinality of 4, i.e., a = 4.The distance between two symbols can be read off by examining the corresponding row and column. For example, *dist(**a**,**b**)* = 0 and *dist(**a**,**c**)* = 0.67.

3. Distance Measures

The value in *dist(r,c)* for any look up table can be calculated by the following expression:

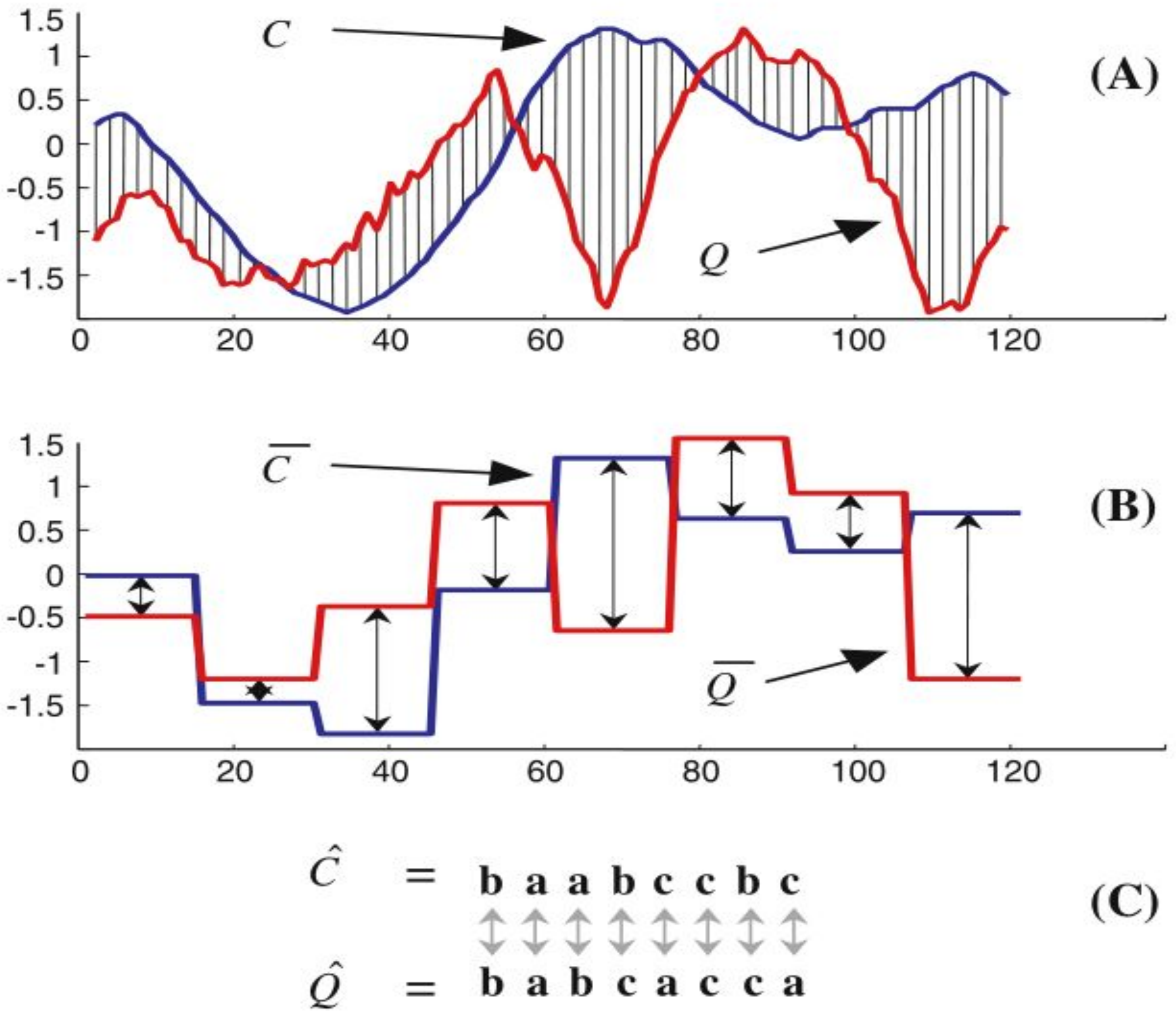
$$dist_{r,c} = \begin{cases} 0, & if\ |r-c| \le 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)}, & otherwoise \end{cases}$$

In the next page, there are a visual intuition of the three distance measured representations. (A) The Euclidean distance between two time series. (B)The distance measure defined for the PAA approximation. (C) The distance between two SAX representations of a time series.

a / βi	3	4	5	6	7	8	9	10
β_1	−0.43	−0.67	−0.84	−0.97	−1.07	−1.15	−1.22	−1.28
β_2	0.43	0	−0.25	−0.43	−0.57	−0.67	−0.76	−0.84
β_3		0.67	0.25	0	−0.18	−0.32	−0.43	−0.52
β_4			0.84	0.43	0.18	0	−0.14	−0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

3. Distance Measures

数据挖掘实验室
Data Mining Lab



Proof:

Step1: We need to show that the PAA distance lower-bounds the Euclidean distance; that is, $D(Q,C) \geq DR(\overline{Q},\overline{C})$

$$\sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \geq \sqrt{\frac{n}{w}}\sqrt{\sum_{i=1}^{w}(\overline{q_i} - \overline{c_i})^2}$$

Let $\overline{Q}$ and $\overline{C}$ be the means of time series Q and C, respectively. Since we are considering only the single-frame case, the above Eq can be written as:

3. Distance Measures

$$\sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \geq \sqrt{n}\sqrt{(\overline{Q} - \overline{C})^2}$$

Squaring both sides we get:

$$\sum_{i=1}^{n}(q_i - c_i)^2 \geq n(\overline{Q} - \overline{C})^2$$

3. Distance Measures

Each point q_i in Q can be represented in term of $\bar{Q}$,i.e. $q_i = \bar{Q} - \Delta q_i$.Same applies to each point c_i in C.

$$\sum_{i=1}^{n}((\overline{Q} - \Delta q_i) - (\overline{C} - \Delta c_i))^2 \geq n(\overline{Q} - \overline{C})^2$$

$$\sum_{i=1}^{n}((\overline{Q} - \overline{C}) - (\Delta q_i - \Delta c_i))^2 \geq n(\overline{Q} - \overline{C})^2$$

3. Distance Measures

数据挖掘实验室
Data Mining Lab

Expand and rewrite:

$$\sum_{i=1}^{n}((\overline{Q}-\overline{C})^2-2(\overline{Q}-\overline{C})(\Delta q_i-\Delta c_i)+(\Delta q_i-\Delta c_i)^2)\geq n(\overline{Q}-\overline{C})^2$$

Then:

$$\sum_{i=1}^{n}(\overline{Q}-\overline{C})^2-\sum_{i=1}^{n}2(\overline{Q}-\overline{C})(\Delta q_i-\Delta c_i)+\sum_{i=1}^{n}(\Delta q_i-\Delta c_i)^2\geq n(\overline{Q}-\overline{C})^2$$

Or:

$$n(\overline{Q}-\overline{C})^2-2(\overline{Q}-\overline{C})\sum_{i=1}^{n}(\Delta q_i-\Delta c_i)+\sum_{i=1}^{n}(\Delta q_i-\Delta c_i)^2\geq n(\overline{Q}-\overline{C})^2$$

3. Distance Measures

数据挖掘实验室

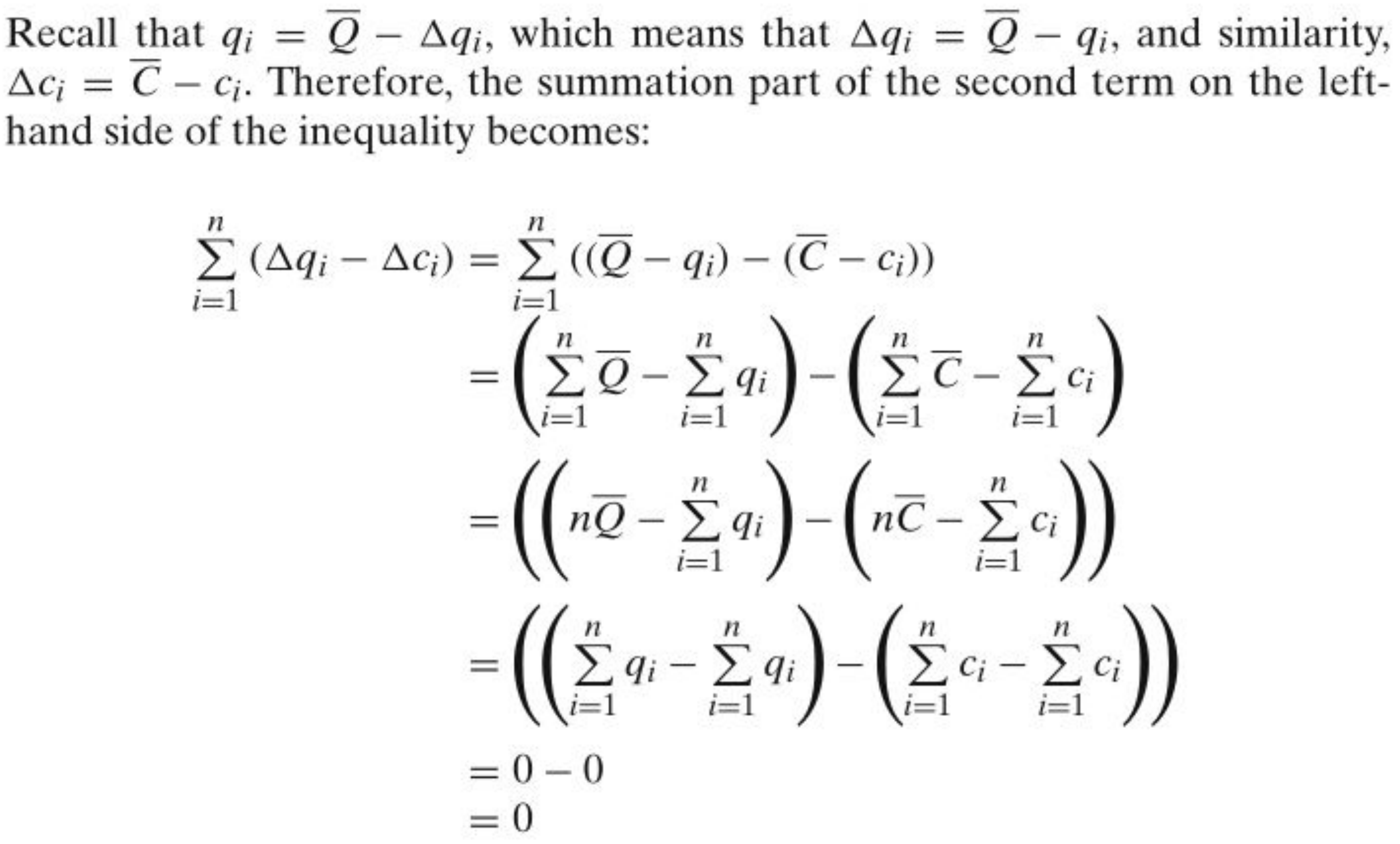
Data Mining Lab

LESS IS MORE

Recall that $q_i = \overline{Q} - \Delta q_i$, which means that $\Delta q_i = \overline{Q} - q_i$, and similarity, $\Delta c_i = \overline{C} - c_i$. Therefore, the summation part of the second term on the left-hand side of the inequality becomes:

$$\begin{aligned}\sum_{i=1}^{n} (\Delta q_i - \Delta c_i) &= \sum_{i=1}^{n} ((\overline{Q} - q_i) - (\overline{C} - c_i)) \\ &= \left(\sum_{i=1}^{n} \overline{Q} - \sum_{i=1}^{n} q_i\right) - \left(\sum_{i=1}^{n} \overline{C} - \sum_{i=1}^{n} c_i\right) \\ &= \left(\left(n\overline{Q} - \sum_{i=1}^{n} q_i\right) - \left(n\overline{C} - \sum_{i=1}^{n} c_i\right)\right) \\ &= \left(\left(\sum_{i=1}^{n} q_i - \sum_{i=1}^{n} q_i\right) - \left(\sum_{i=1}^{n} c_i - \sum_{i=1}^{n} c_i\right)\right) \\ &= 0 - 0 \\ &= 0\end{aligned}$$

3. Distance Measures



Recall that $q_i = \overline{Q} - \Delta q_i$, which means that $\Delta q_i = \overline{Q} - q_i$, and similarity, $\Delta c_i = \overline{C} - c_i$. Therefore, the summation part of the second term on the left-hand side of the inequality becomes:

$$\begin{aligned}
\sum_{i=1}^{n} (\Delta q_i - \Delta c_i) &= \sum_{i=1}^{n} ((\overline{Q} - q_i) - (\overline{C} - c_i)) \\
&= \left(\sum_{i=1}^{n} \overline{Q} - \sum_{i=1}^{n} q_i\right) - \left(\sum_{i=1}^{n} \overline{C} - \sum_{i=1}^{n} c_i\right) \\
&= \left(\left(n\overline{Q} - \sum_{i=1}^{n} q_i\right) - \left(n\overline{C} - \sum_{i=1}^{n} c_i\right)\right) \\
&= \left(\left(\sum_{i=1}^{n} q_i - \sum_{i=1}^{n} q_i\right) - \left(\sum_{i=1}^{n} c_i - \sum_{i=1}^{n} c_i\right)\right) \\
&= 0 - 0 \\
&= 0
\end{aligned}$$

3. Distance Measures

Substituting 0 into the Eq. Becomes:

$$n(\overline{Q}-\overline{C})^2 - 0 + \sum_{i=1}^{n}(\Delta q_i - \Delta c_i)^2 \geq n(\overline{Q}-\overline{C})^2$$

$$\Downarrow$$

$$\sum_{i=1}^{n}(\Delta q_i - \Delta c_i)^2 \geq 0$$

Which always holds true, hence completes the proof.

- Data mining problems are I/O bound

- The generic data mining algorithm mitigates the problem, if you can obey the lower bounding requirement.

- There is one approximation of time series that is symbolic and lower bounding, SAX

- What’s the advantage of SAX in practice ?

Hierarchical Cluster

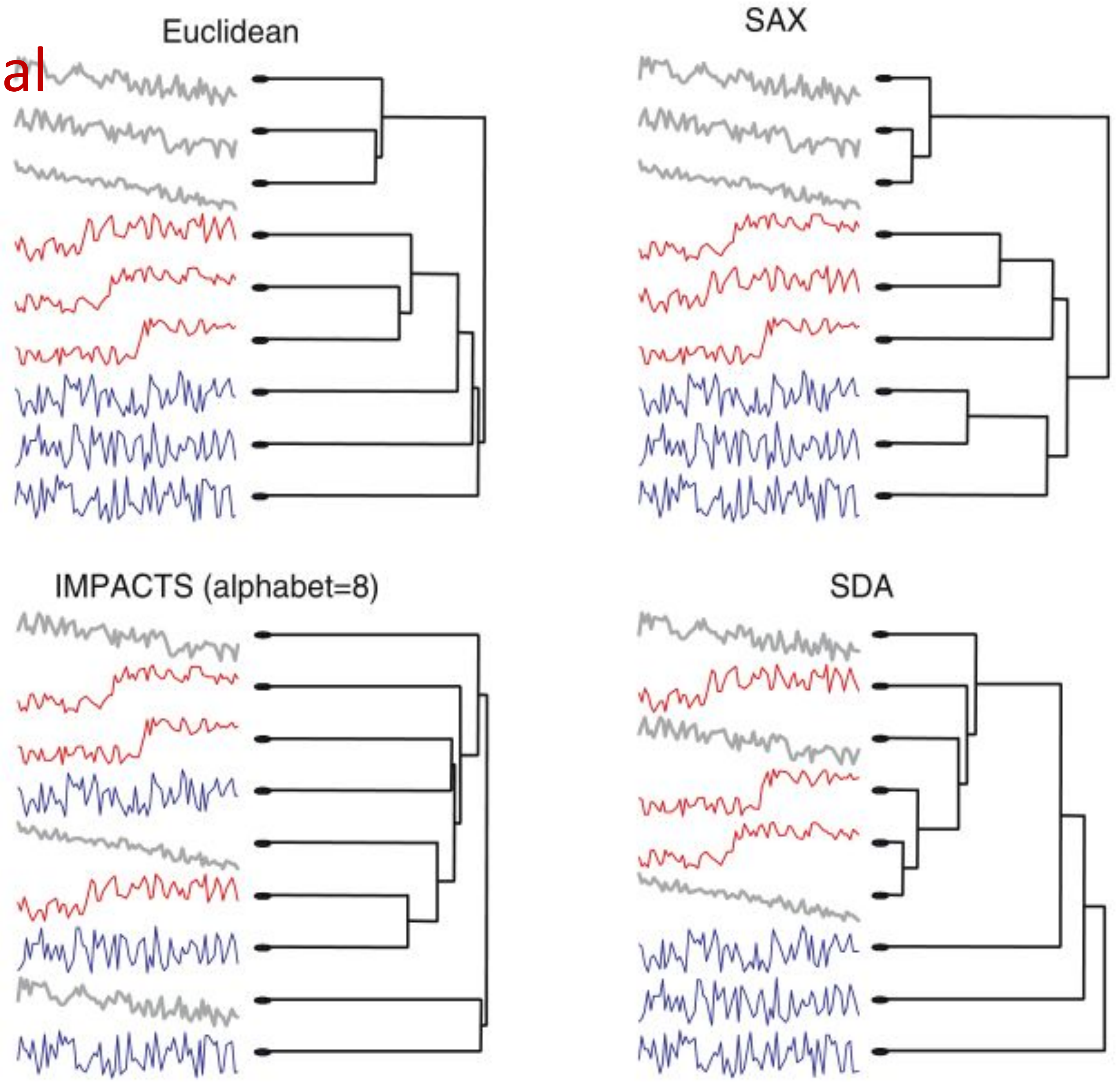
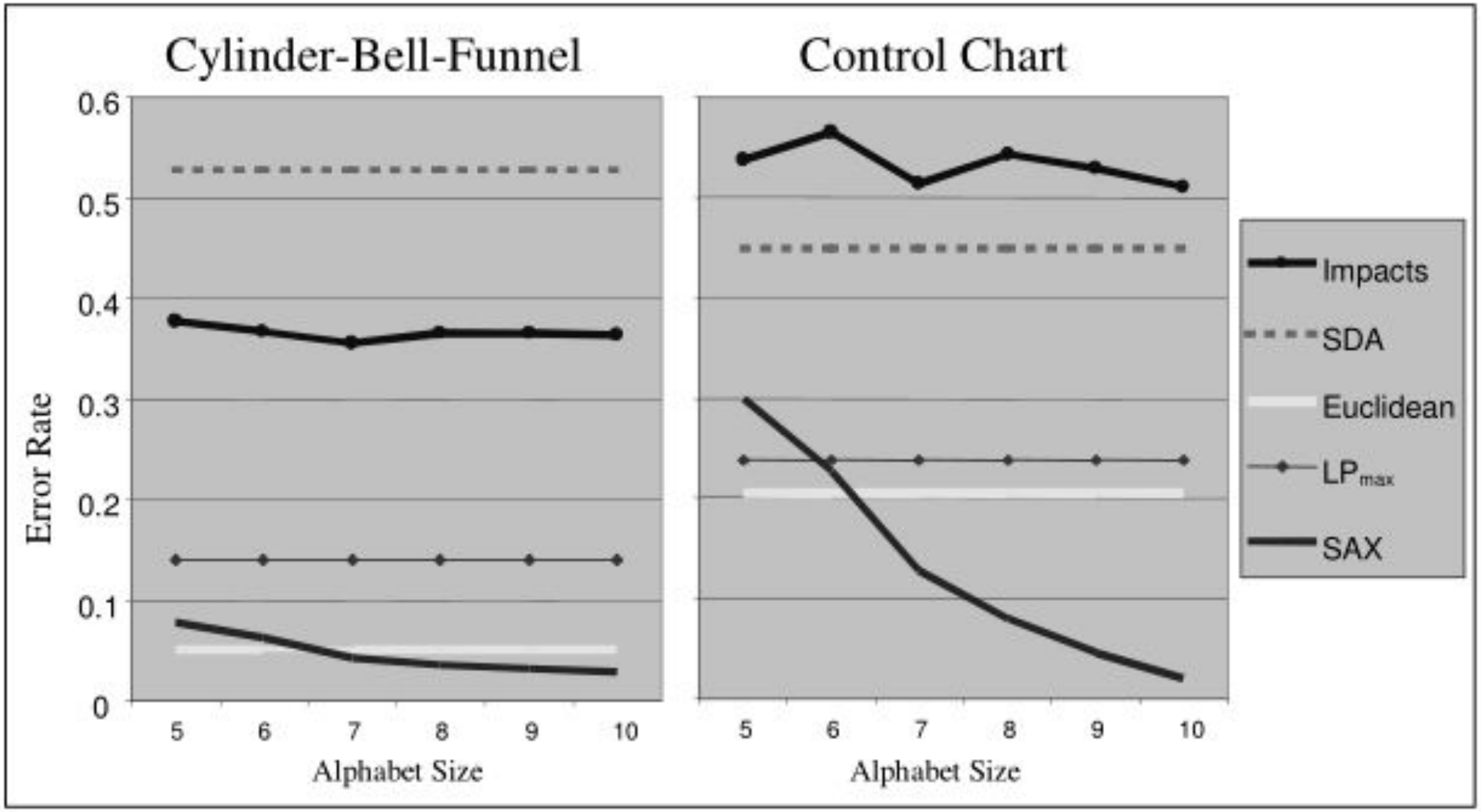


Fig. 12 A comparison of the four distance measures' ability to cluster members of the Control Chart dataset. Complete linkage was used as the agglomeration technique

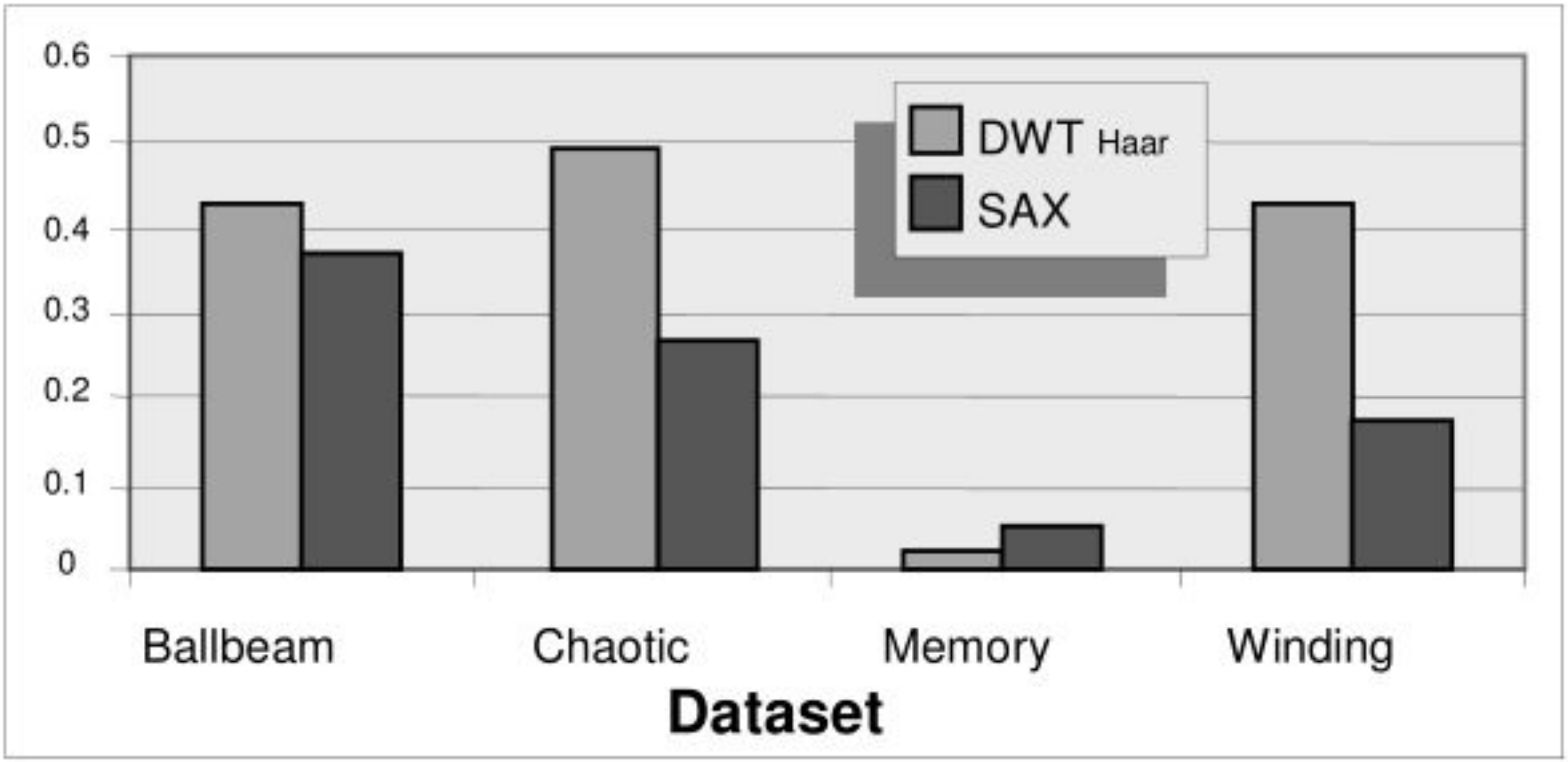
K-NN Classification(50 datasets)



A comparison of five distance measures utility for nearest neighbor classification. We tested different alphabet sizes for SAX and IMPACTS. SDA's alphabet size is fixed at 5.

Query

数据挖掘实验室
Data Mining Lab

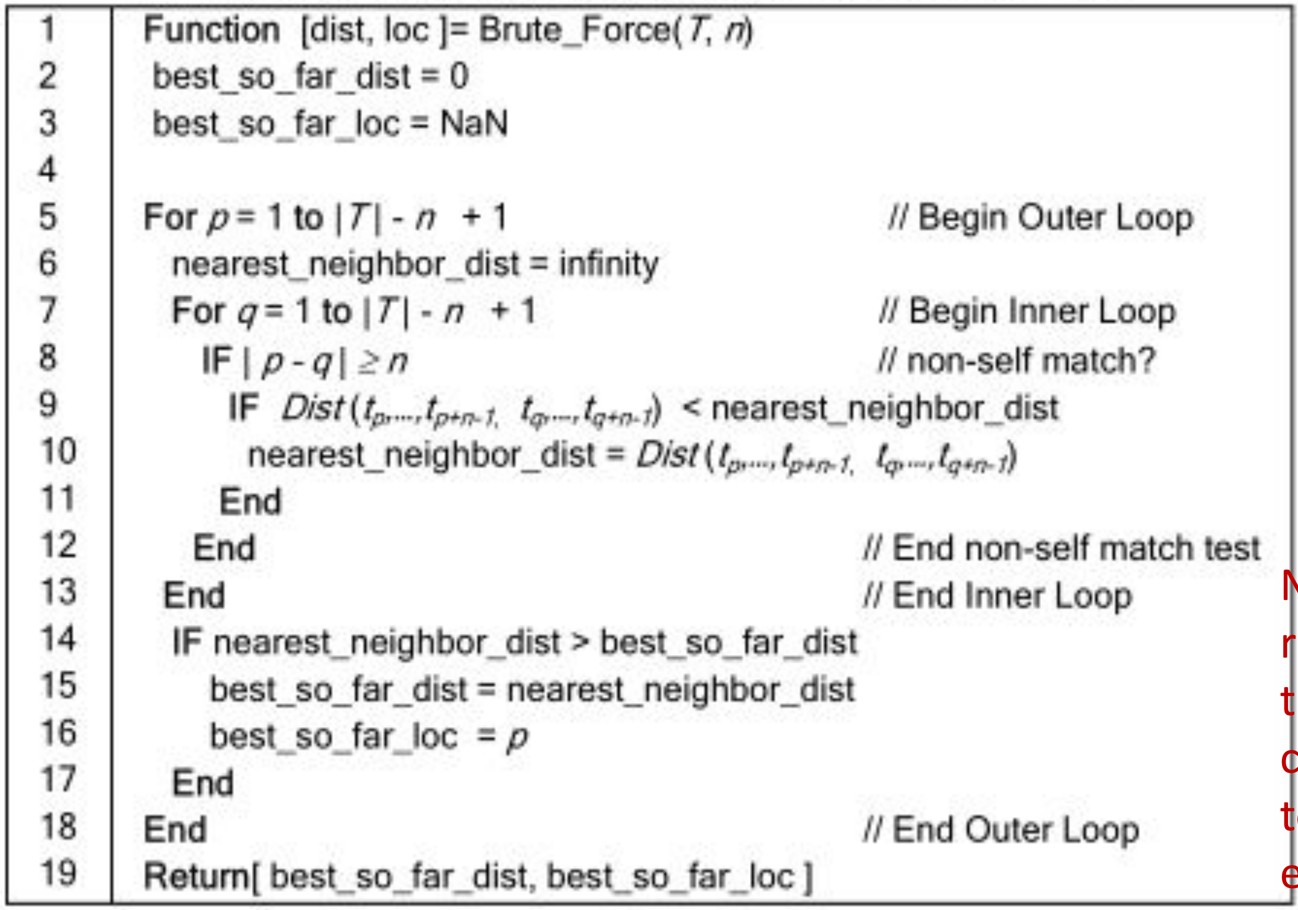


A comparison of indexing ability of wavelets versus SAX. The Y-axis is the percentage of the data that must be retrieved from disk to answer a 1-NN query of length 256

The Utility of SAX ?

4. Using SAX to Find time series Discords

Table 1: Brute Force Discord Discovery.

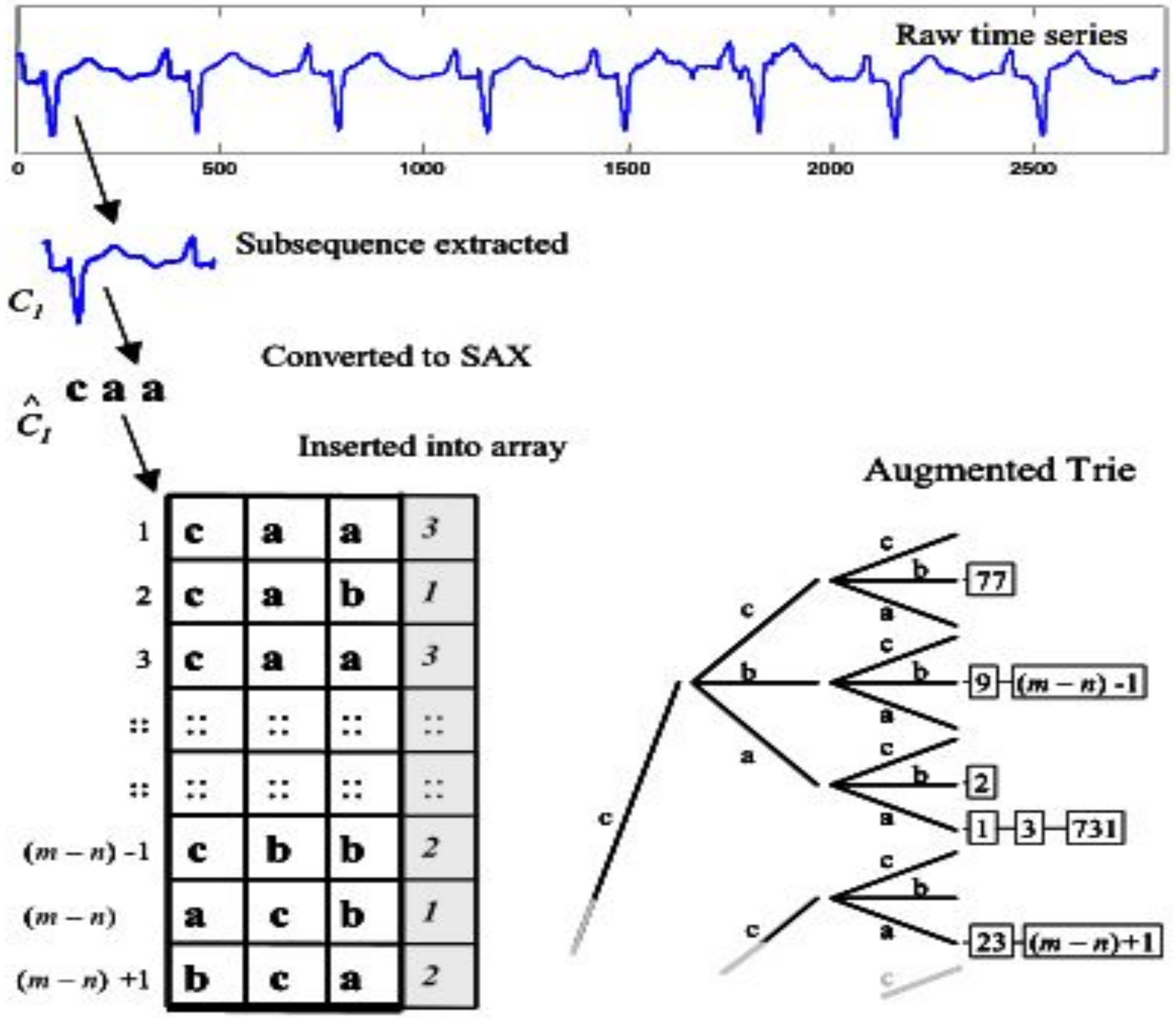


```
Function [dist, loc ]= Brute_Force($T$, $n$)
 best_so_far_dist = 0
 best_so_far_loc = NaN

For $p$ = 1 to $|T| - n + 1$                    // Begin Outer Loop
  nearest_neighbor_dist = infinity
  For $q$ = 1 to $|T| - n + 1$                  // Begin Inner Loop
    IF $|p - q| \geq n$                          // non-self match?
      IF $Dist(t_p,\dots,t_{p+n-1},\ t_q,\dots,t_{q+n-1})$ < nearest_neighbor_dist
        nearest_neighbor_dist = $Dist(t_p,\dots,t_{p+n-1},\ t_q,\dots,t_{q+n-1})$
      End
    End                                          // End non-self match test
  End                                            // End Inner Loop
  IF nearest_neighbor_dist > best_so_far_dist
     best_so_far_dist = nearest_neighbor_dist
     best_so_far_loc = $p$
  End
End                                              // End Outer Loop
Return[ best_so_far_dist, best_so_far_loc ]
```

Note that the algorithm requires exactly one parameter, the length of subsequences to consider. The algorithm is easy to implement and produces exact results. However, it has one fatal flaw for data mining. It has $O(m^2)$ time complexity which is simply untenable for even moderately large datasets.

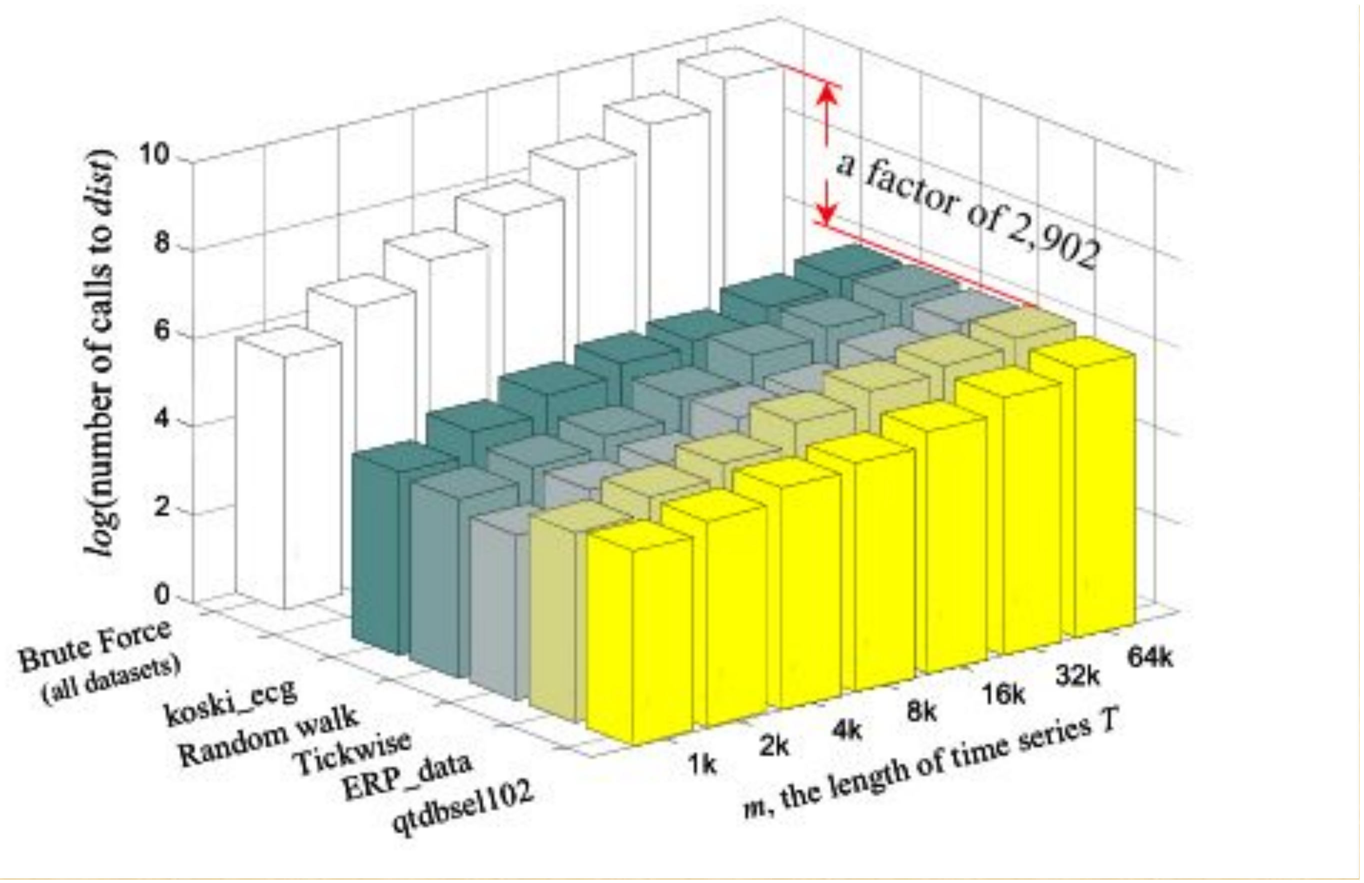
4. Using SAX to Find time series Discords



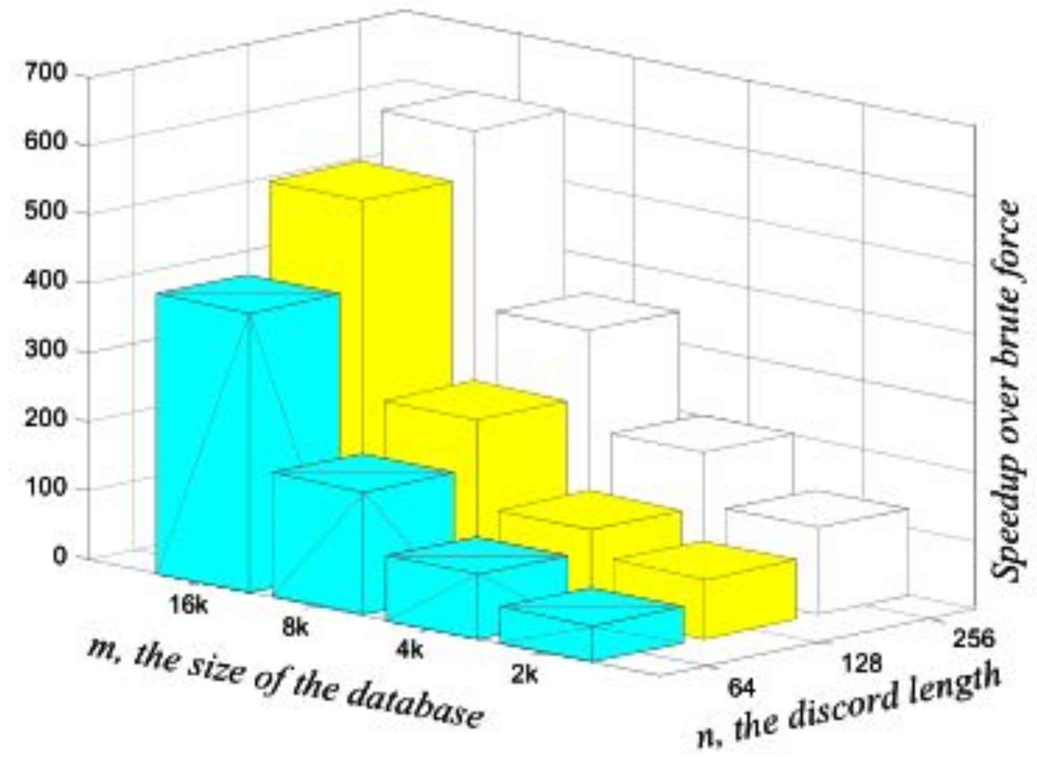
Both data structures can be created in time and space linear in the length of $T[1]$.

Only need $[log_2(\alpha)]$ bits for each SAX symbols, both data structures are significantly smaller than the raw time series data

Time complexity is $T[n]$



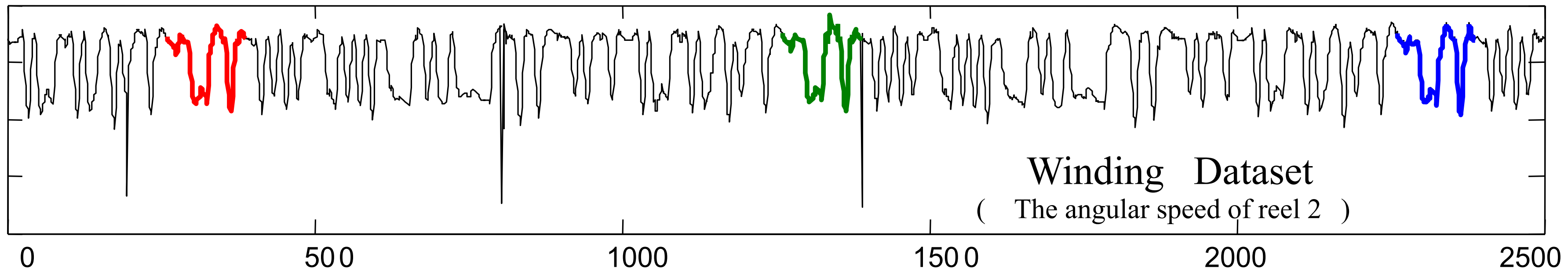
Note that as the data sizes increase, the differences get larger. For a time series of length 64,000, SAX is almost three thousand times faster than brute force for all datasets.

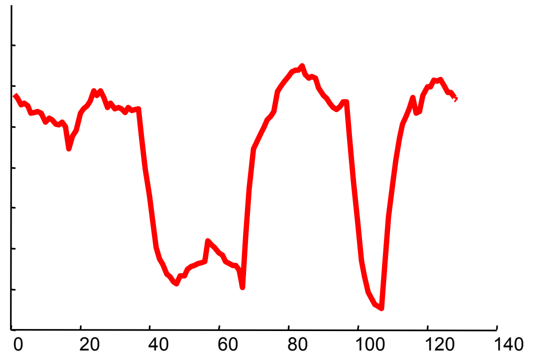


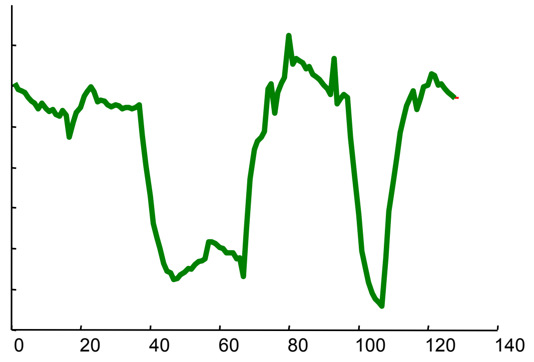
Length of at least 16,000, 82 different dataset

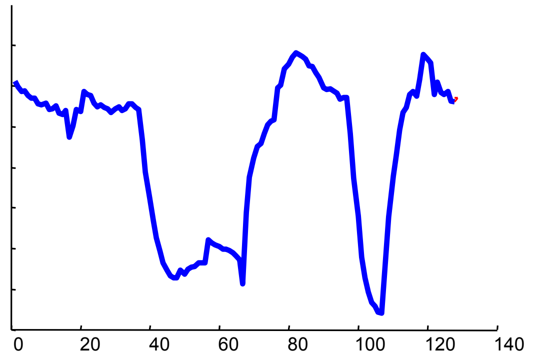
Time Series Motif Discovery

(finding repeated patterns)

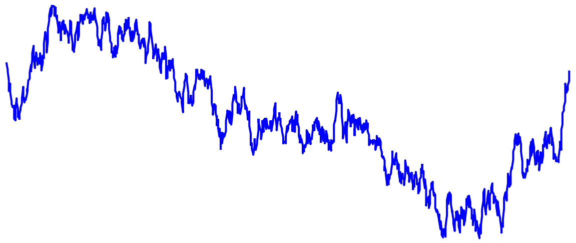


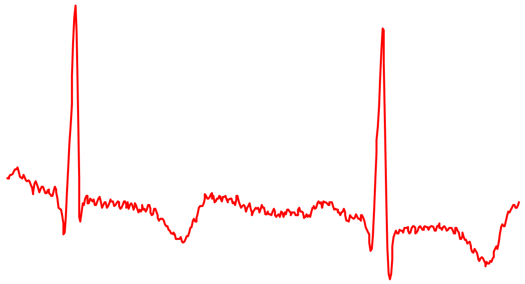






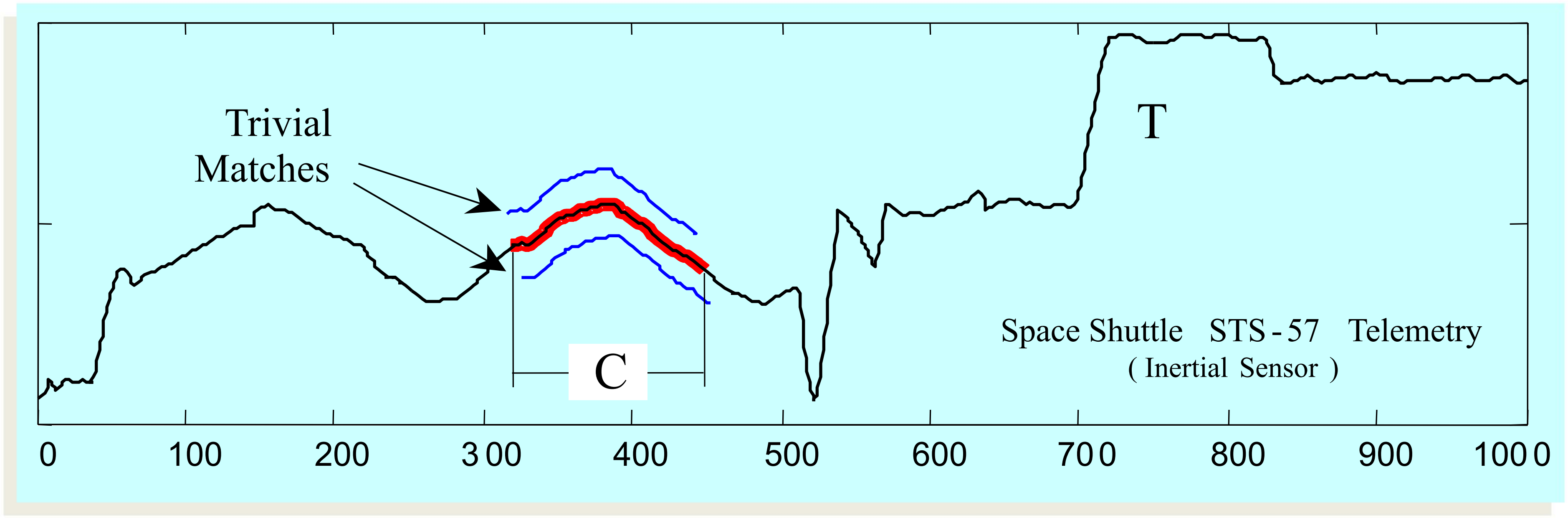
Why Find Motifs?


River Stream


Stocks

CCGTGCTAGGCCCCACCCCTACC
TTGCAGTCCCCGCAAGCTCATCT
GCGCGAACCGGAACGCCCACCA
CCCTTGGGTTGAAATTAAGGAG
GCGGTTGGCAGCTTCCGAGGCG
CACGTACCTGCGAATAAATAACT
GTCCGC

Gene



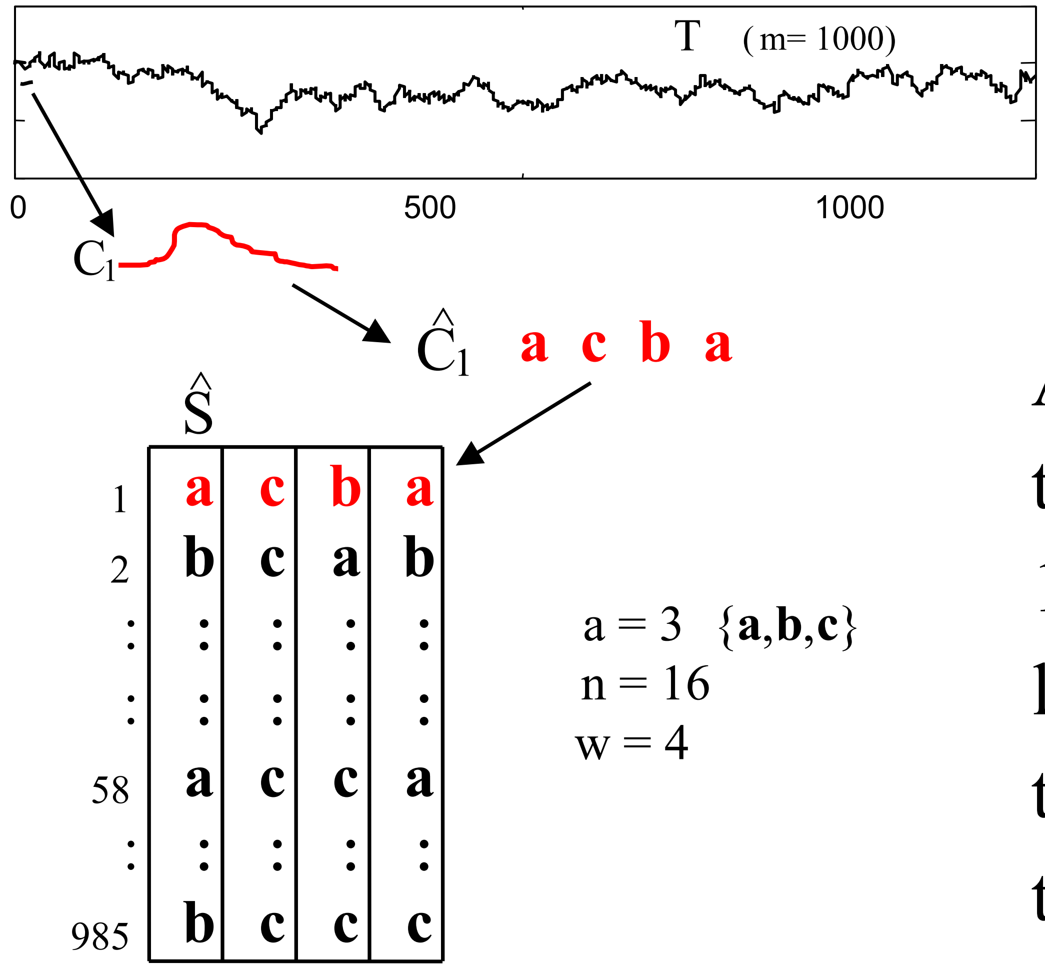
Definition 1. Match: Given a positive real number R (called range) and a time series T containing a subsequence C beginning at position p and a subsequence M beginning at q, if $D(C, M) \leq R$, then M is called a matching subsequence of C.

Definition 2. Trivial Match: Given a time series T, containing a subsequence C beginning at position p and a matching subsequence M beginning at q, we say that M is a trivial match to C if either $p = q$ or there does not exist a subsequence M' beginning at q' such that $D(C, M') > R$, and either $q < q' < p$ or $p < q' < q$.

Definition 3. K-Motif(n,R): Given a time series T, a subsequence length n and a range R, the most significant motif in T (hereafter called the 1-Motif(n,R)) is the subsequence C_1 that has highest count of non-trivial matches (ties are broken by choosing the motif whose matches have the lower variance). The K^{th} most significant motif in T (hereafter called the K-Motif(n,R)) is the subsequence C_K that has the highest count of non-trivial matches, and satisfies $D(C_K, C_i) > 2R$, for all $1 \leq i < K$.

The most reference algorithm is based on a hot idea from bioinformatics, *random projection** and the fact that SAX allows use to ***lower bound*** discrete representations of time series.

A simple worked example of the motif discovery algorithm



1	a	c	b	a
2	b	c	a	b
:	:	:	:	:
:	:	:	:	:
58	a	c	c	a
:	:	:	:	:
985	b	c	c	c

$a = 3$ {**a,b,c**}
$n = 16$
$w = 4$

Assume that we have a time series T of length 1,000, and a motif of length 16, which occurs twice, at time T_1 and time T_{58}.

Using Random Projection is reasonable:

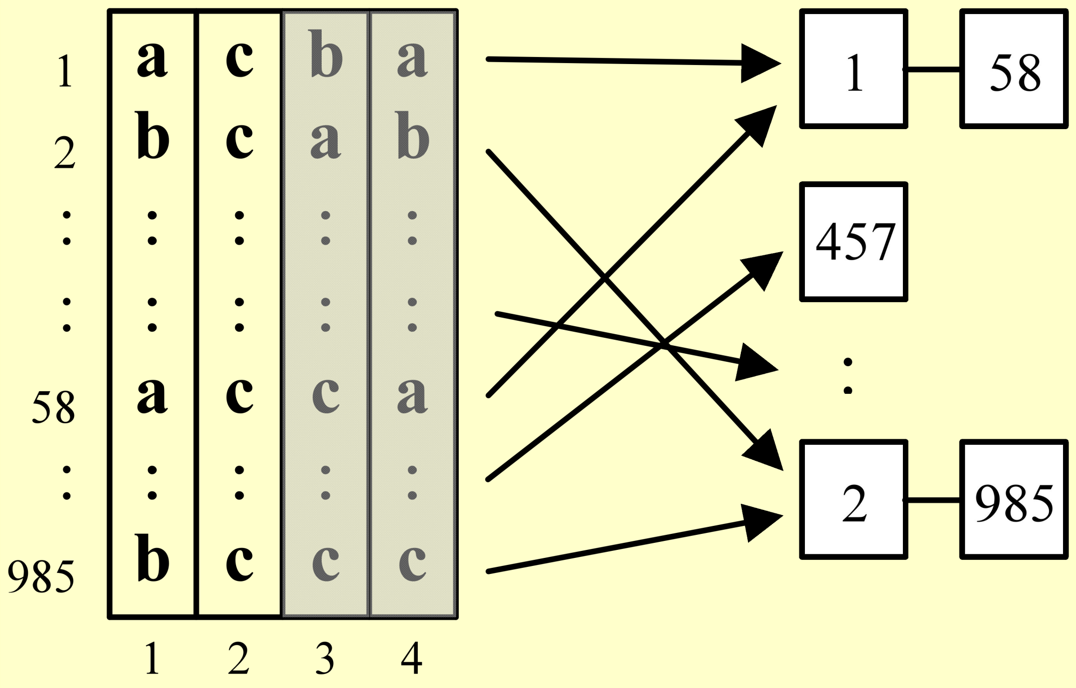
It guarantees that a set of random words over the alphabet will be equiprobability compared.

Applying it to larger subsequences would practically require all pairwise subsequence comparisons to be performed.

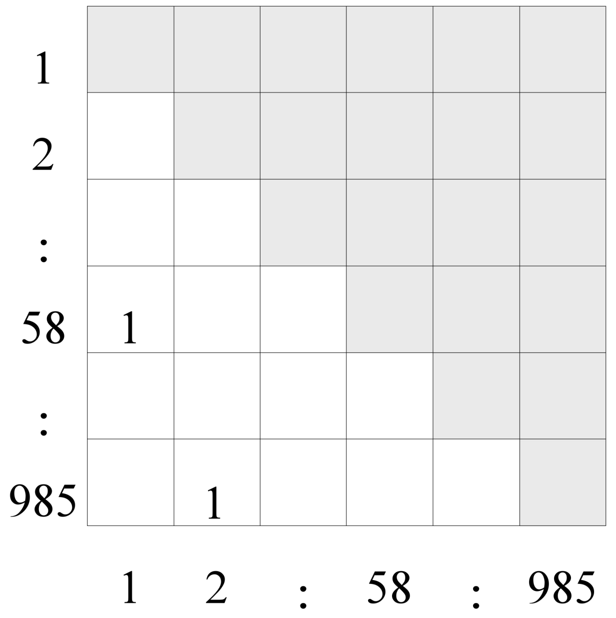
Random Projection

数据挖掘实验室
Data Mining Lab

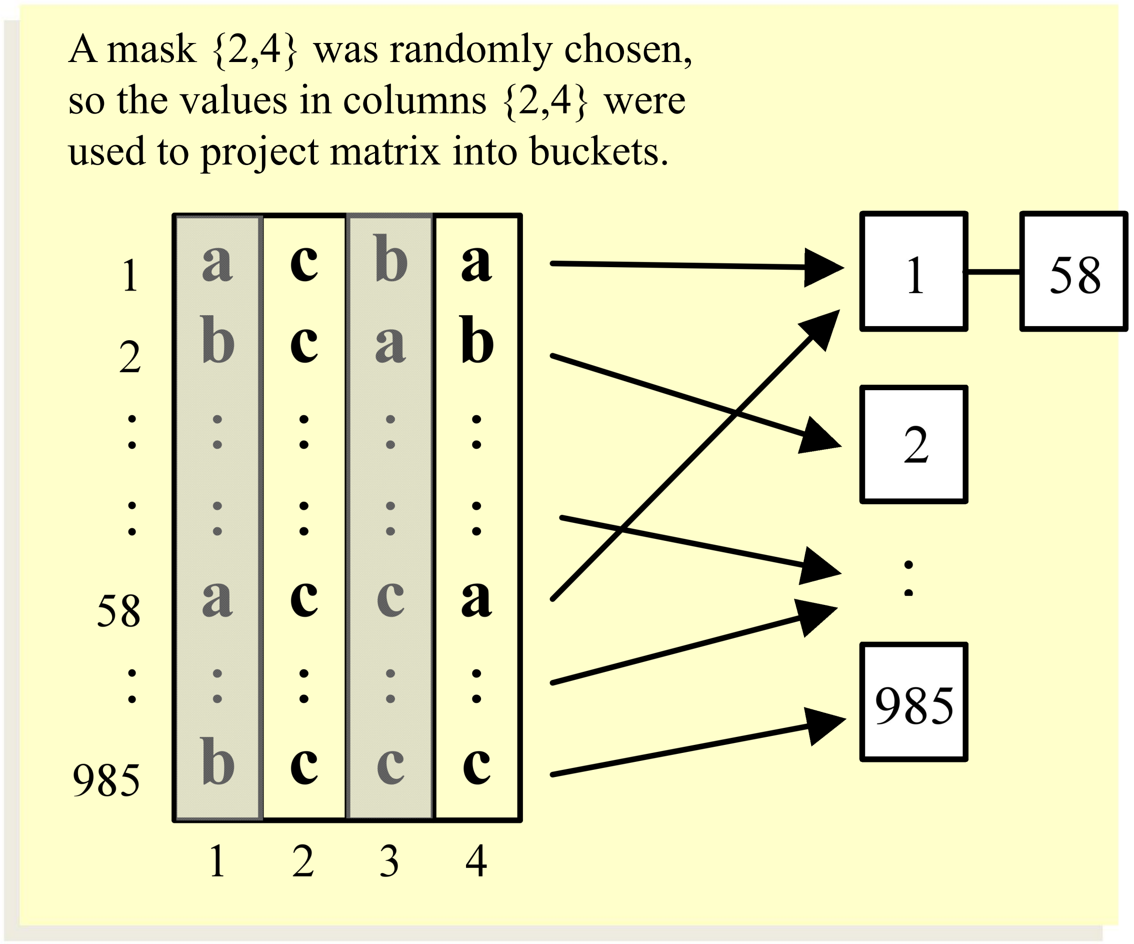
A mask {1,2} was randomly chosen, so the values in columns {1,2} were used to project matrix into buckets.



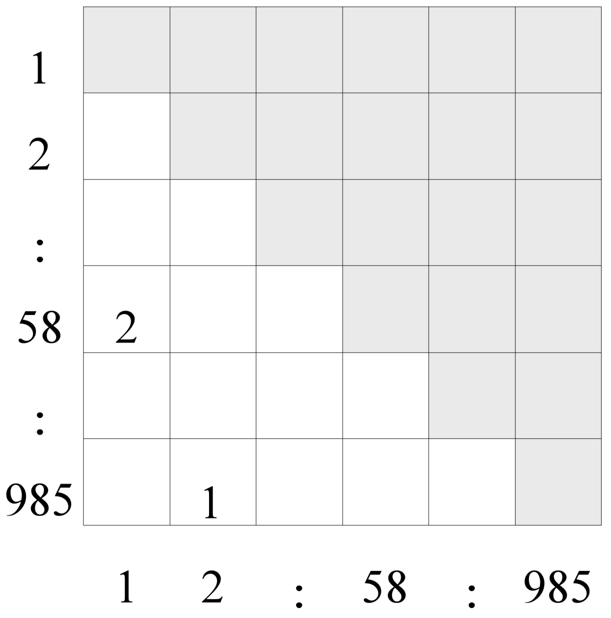
Collisions are recorded by incrementing the appropriate location in the collision matrix



Random Projection

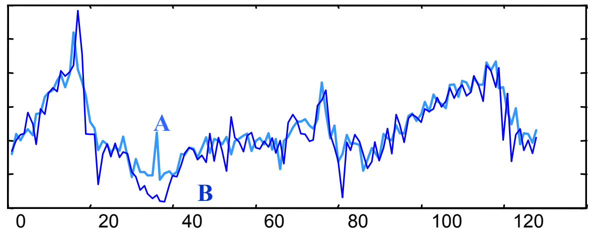

A mask {2,4} was randomly chosen, so the values in columns {2,4} were used to project matrix into buckets.
1
2
58
985
a c b a
b c a b
a c c a
b c c c
1 2 3 4
1
58
2
985

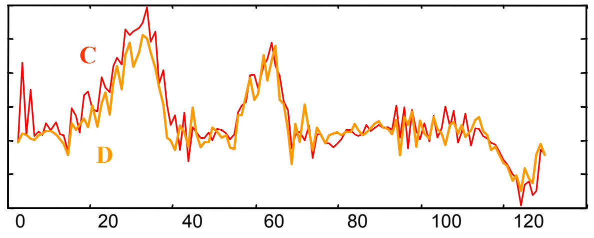
Once again, collisions are recorded by incrementing the appropriate location in the collision matrix

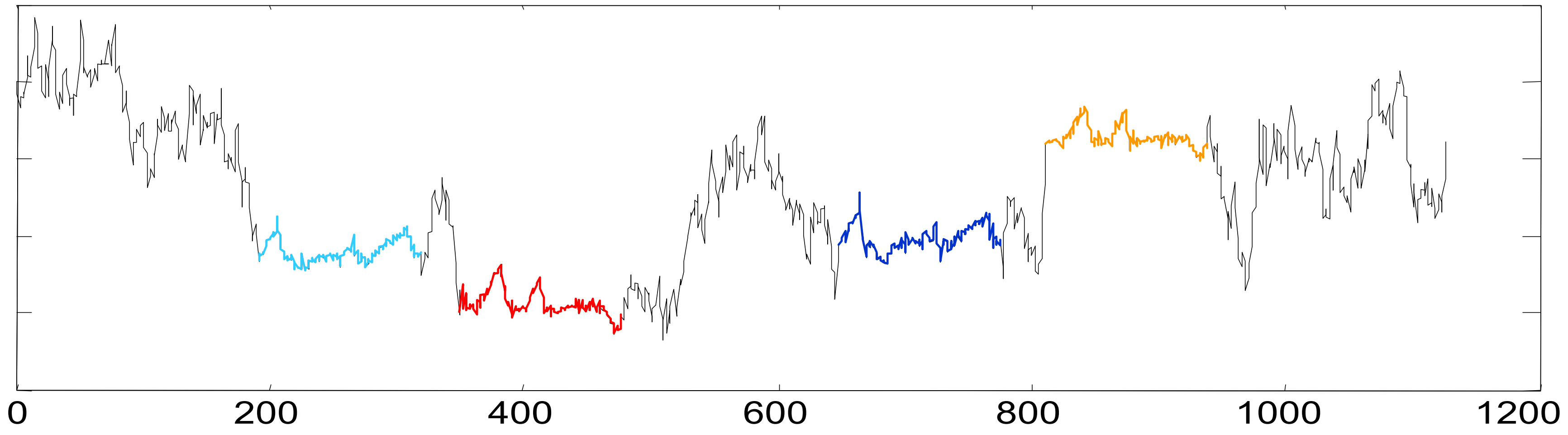


A Simple Experiment

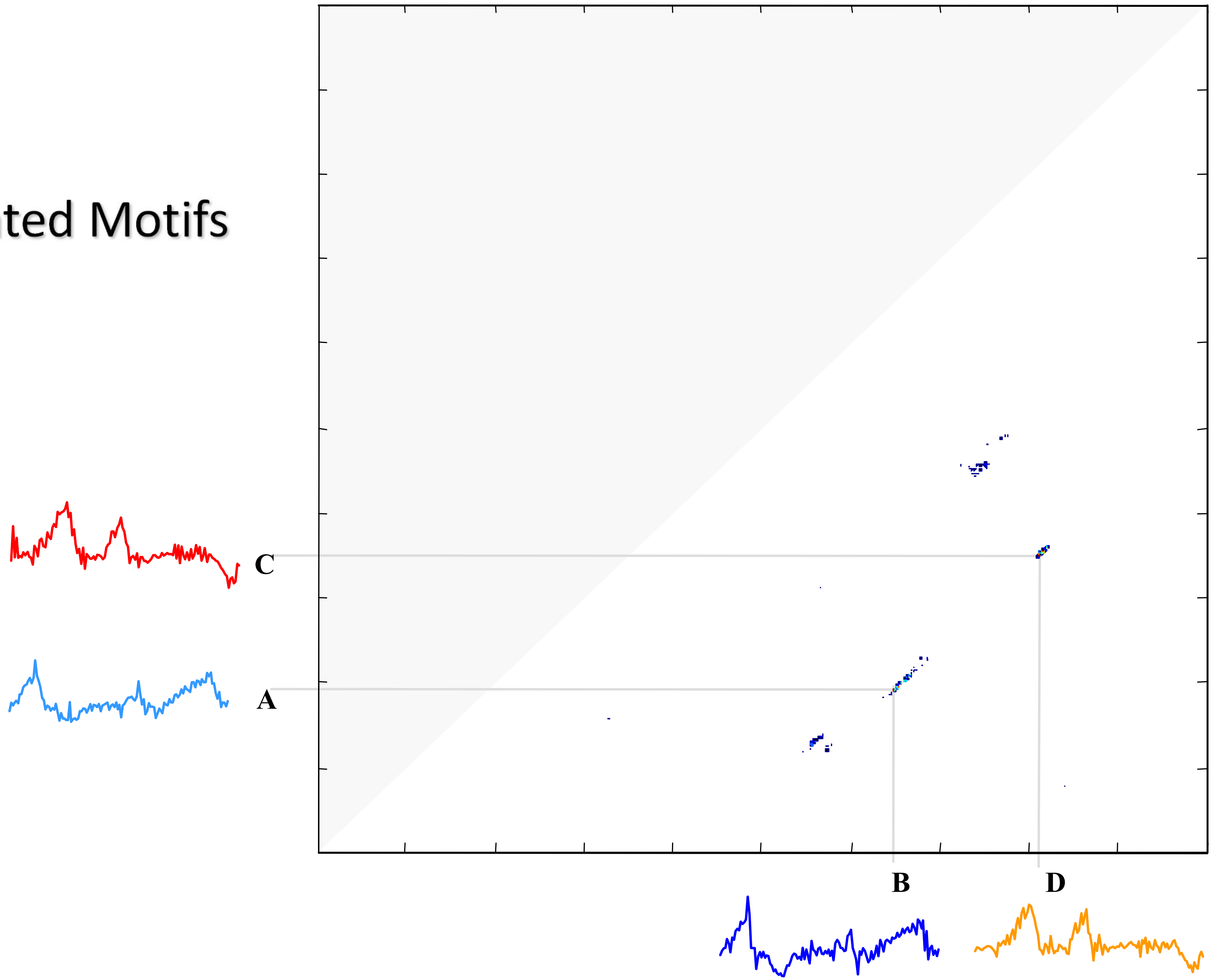
Imbed two motifs into a random walk time series, and see if we can recover them







Planted Motifs



Thanks

Heng Zhang

hengzhang64@gmail.com