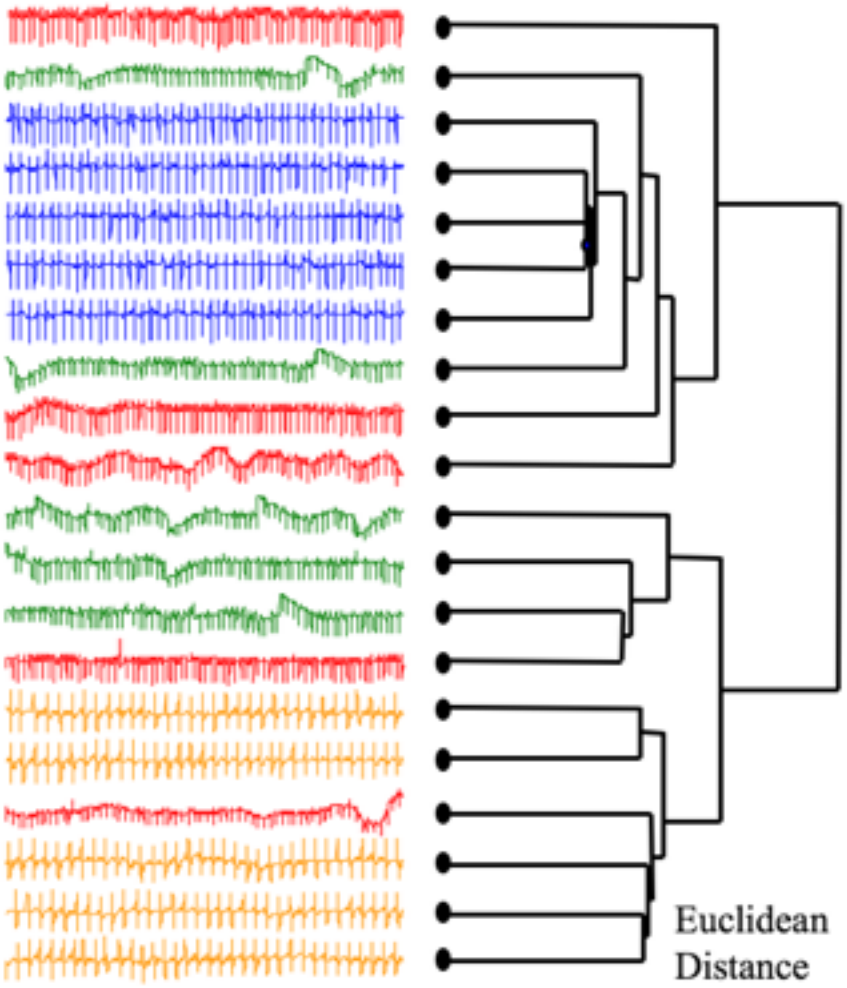


Symbolic Representations

EE382V Activity Sensing and Recognition

UT Austin • Dept. Electrical and Computer Engineering • Fall 2016

Today

Discuss findings from audio sensing activity

Symbolic representations for time series

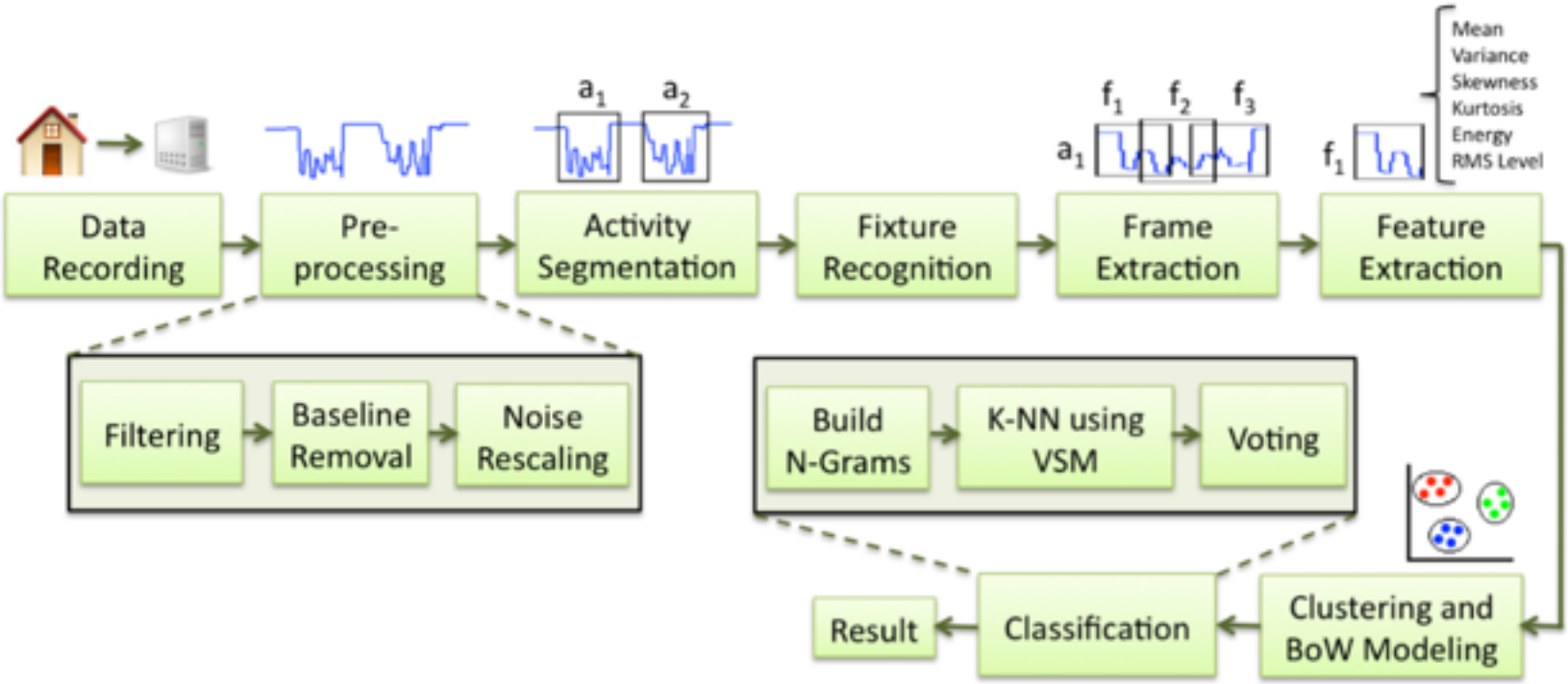
- A new way of thinking about activity recognition problems
- Symbolic Aggregate Approximation (SAX)
- Random Projections

Audio Sensing Lab

Discuss findings from audio sensing activity (by group)

- How did you extract audio features?
- Which features did you choose?
- Which ML algorithms did you try training your classifier with?
- Which resulted in best performance?
- Which technique did you use to evaluate your classifier?
- How did your classifier perform with the test files?
- What did you find most challenging about this exercise?

Vector Space Model for Activity Recognition



Vector Space Model

Documents and queries are represented as vectors.

$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$
$$q = (w_{1,q}, w_{2,q}, \ldots, w_{n,q})$$

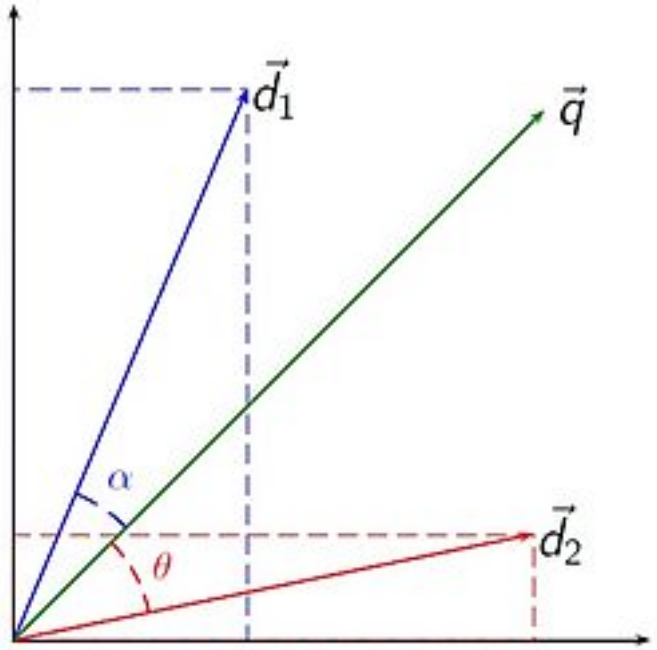
(term) weights,

If a term occurs in the document, its value in the vector is non-zero

$$w_{t,d} = \mathrm{tf}_{t,d} \cdot \log \frac{|D|}{|\{d' \in D \,|\, t \in d'\}|}$$

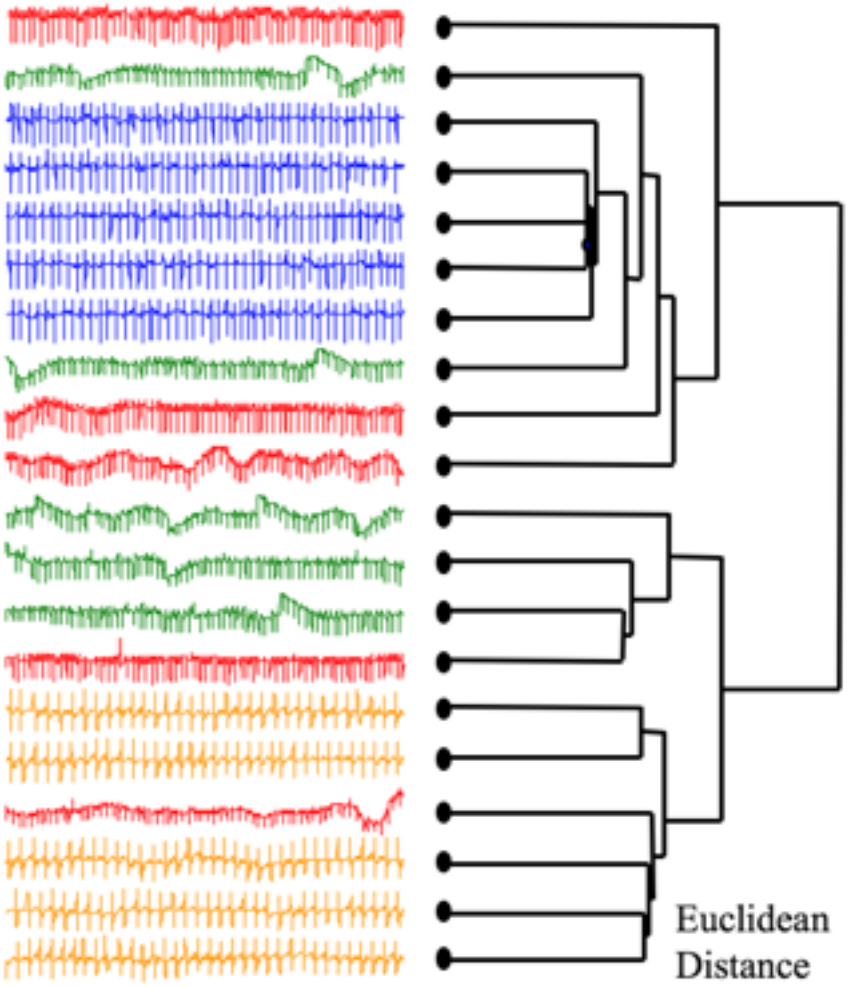
tf-idf weights

Vector Space Model



Relevance rankings of documents in a keyword search can be calculated by comparing the deviation of angles between each document vector and the original query vector

$$\cos\theta = \frac{\mathbf{d_2} \cdot \mathbf{q}}{\|\mathbf{d_2}\| \, \|\mathbf{q}\|}$$



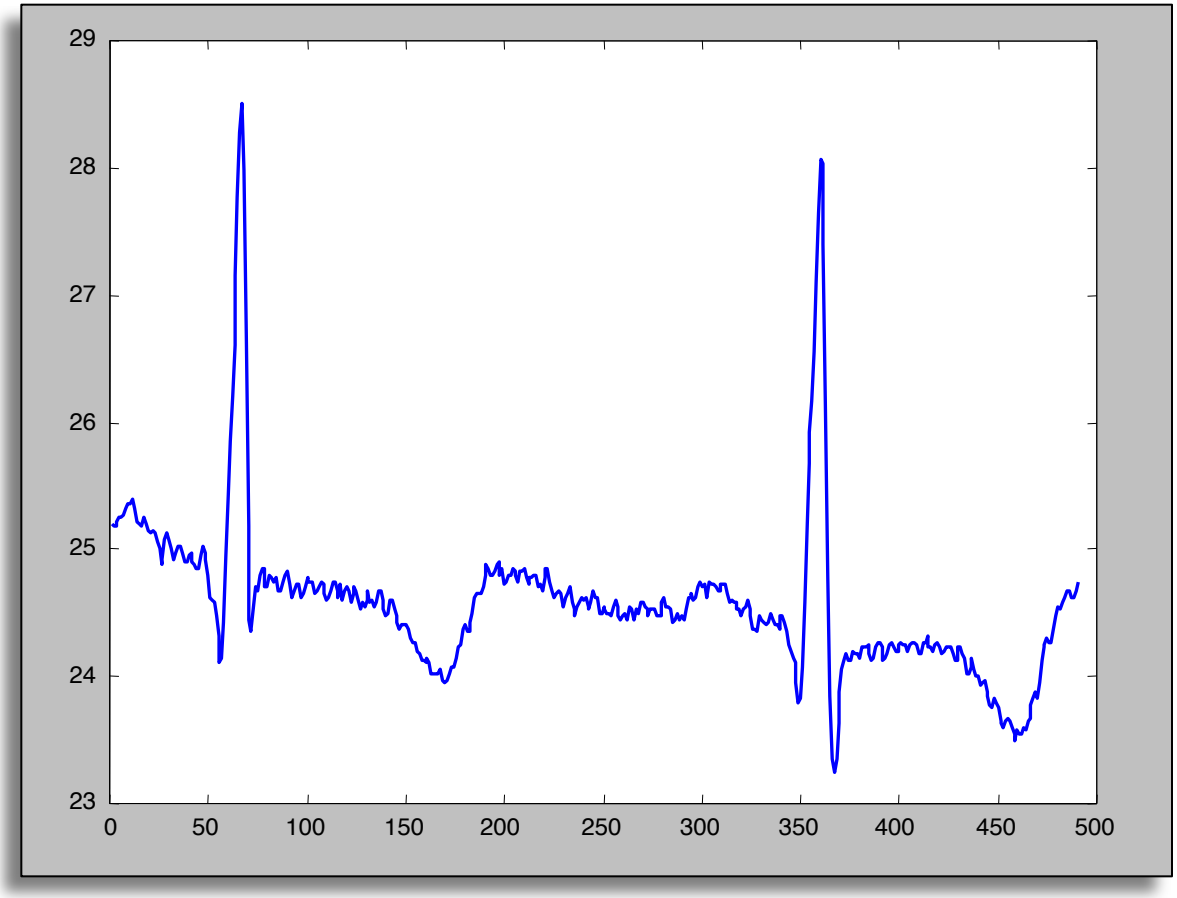
Slides thanks to Eamon Keogh and Jessica Lin

University of California Riverside

What are Time Series?

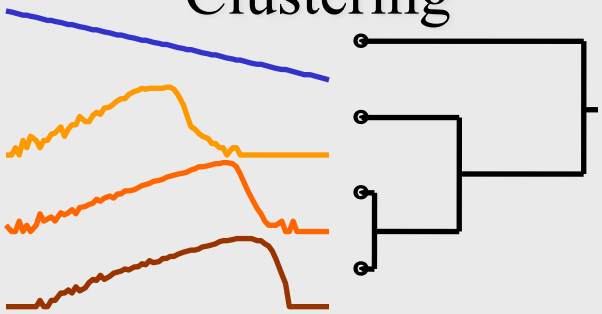
25.1750
25.2250
25.2500
25.2500
25.2750
25.3250
25.3500
25.3500
25.4000
25.4000
25.3250
25.2250
25.2000
25.1750
..
..
24.6250
24.6750
24.6750
24.6250
24.6250
24.6250
24.6750
24.7500

A time series is a collection of observations made sequentially in time.

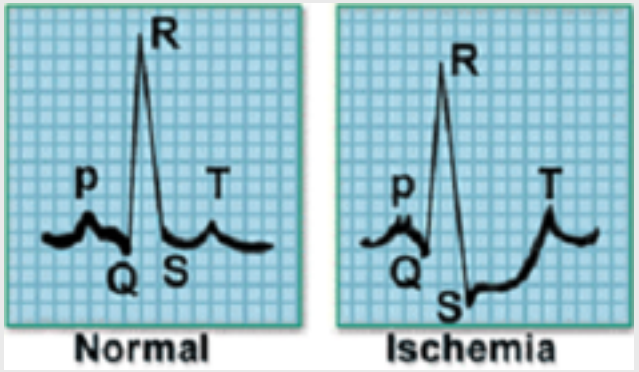


What do we want to do with the time series data?

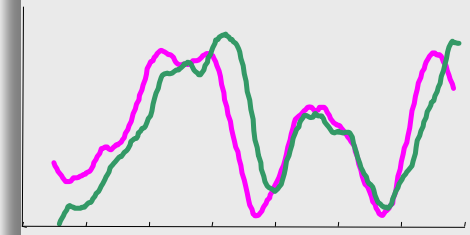
Clustering



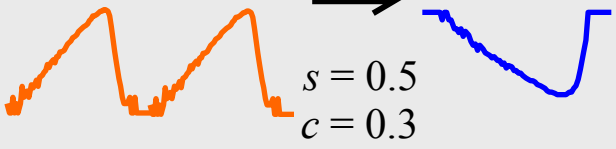
Classification



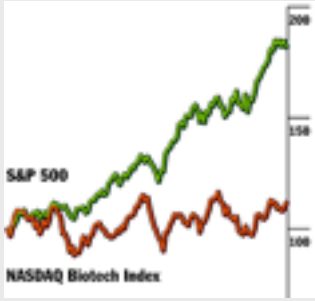
Motif Discovery



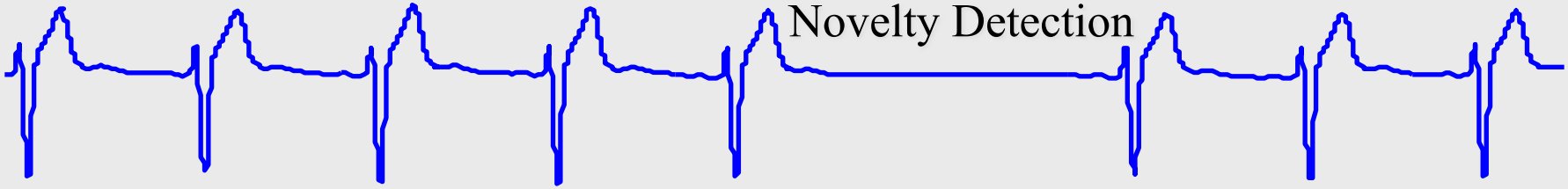
Rule Discovery



Query by Content

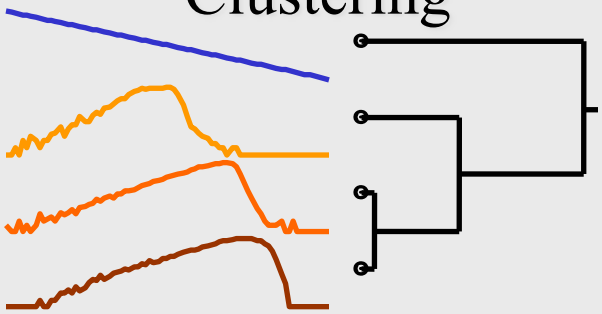


Novelty Detection

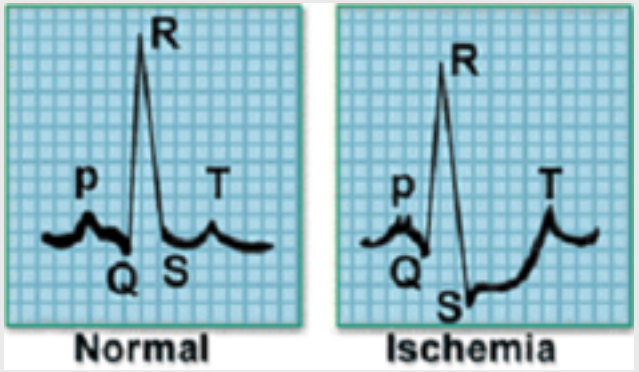


All these problems require **similarity** matching

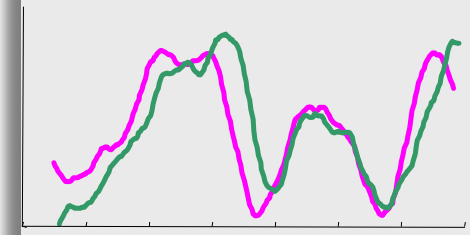
Clustering



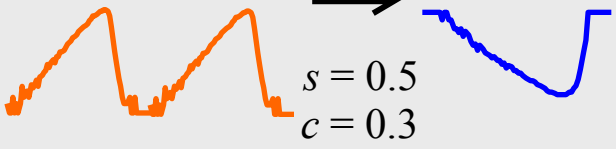
Classification



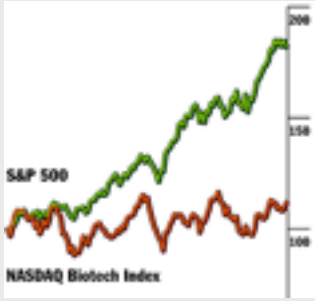
Motif Discovery



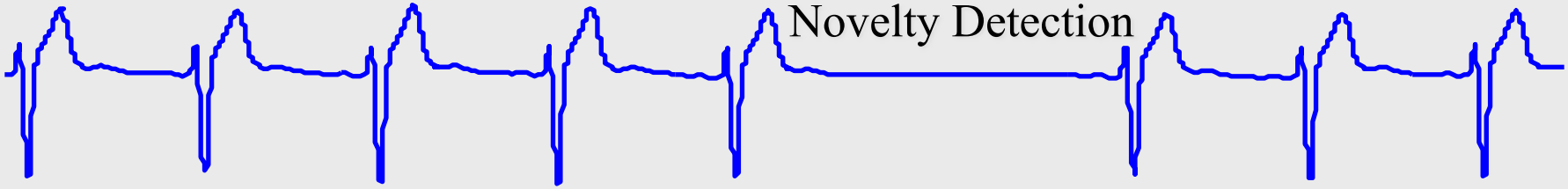
Rule Discovery



Query by Content



Novelty Detection



Euclidean Distance Metric

Given two time series

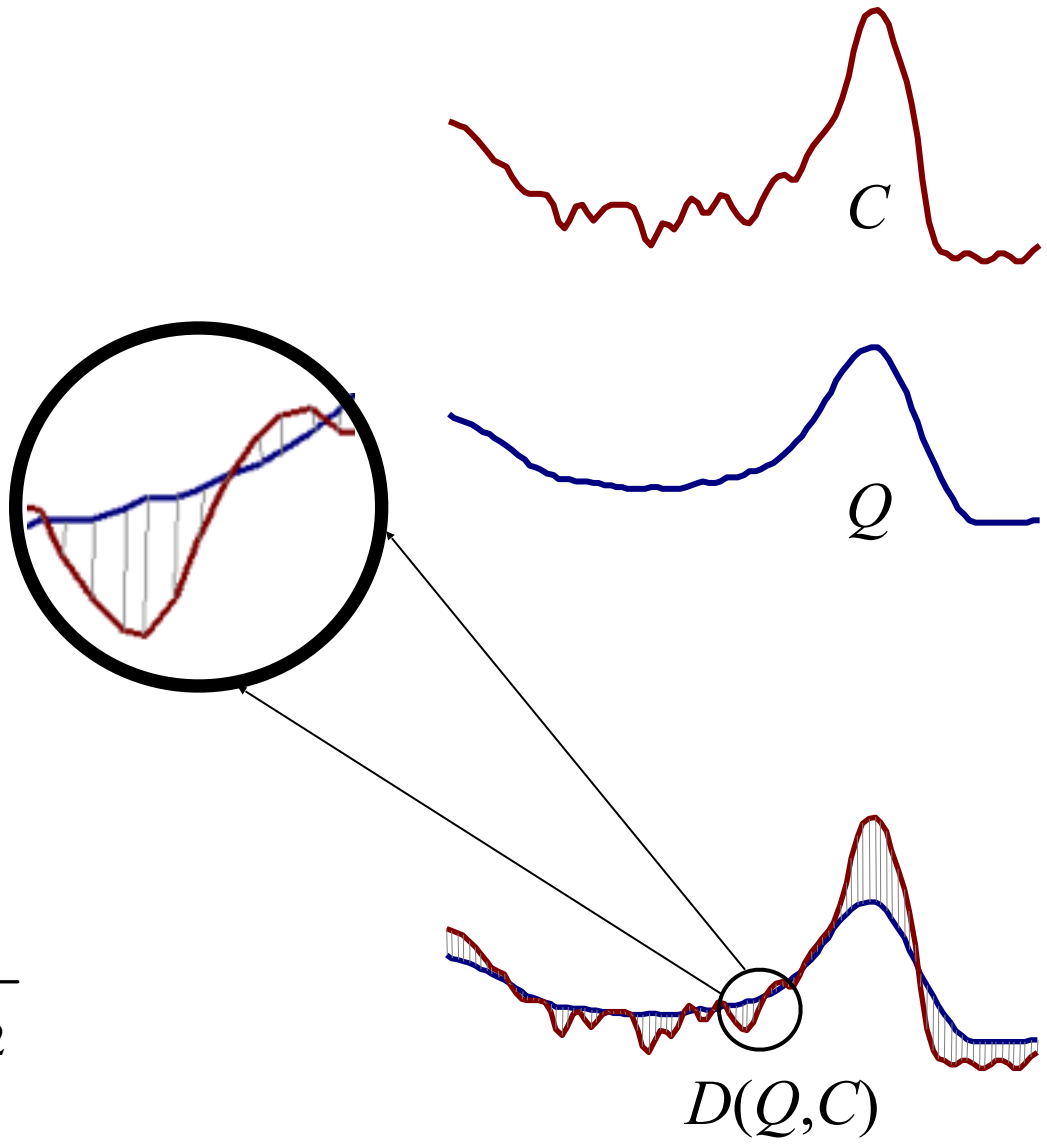
$$Q = q_1 \ldots q_n$$

and

$$C = c_1 \ldots c_n$$

their Euclidean distance is defined as:

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

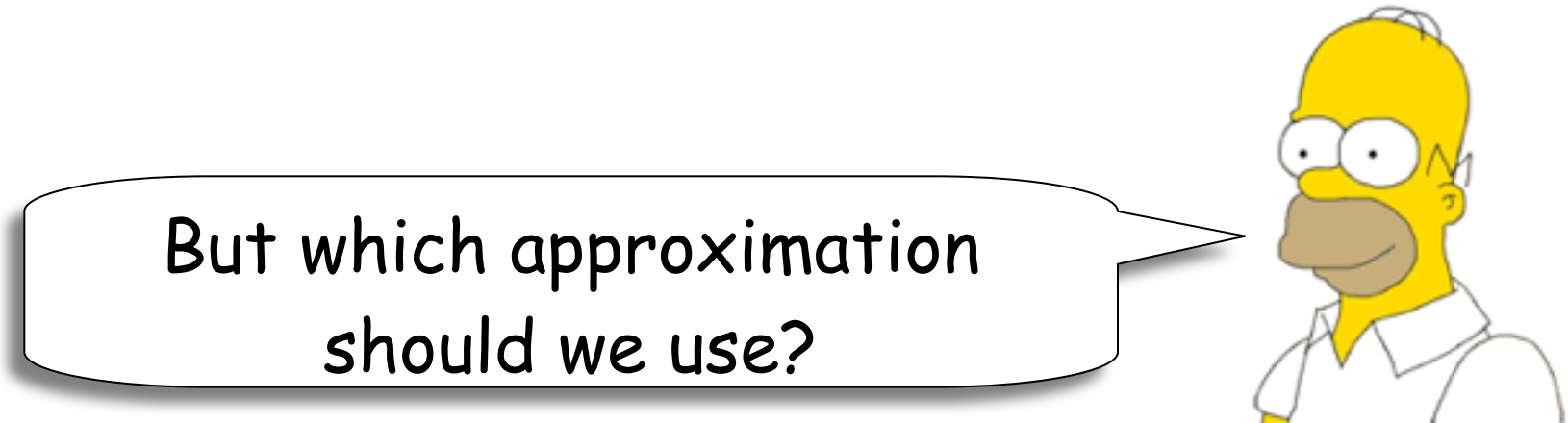


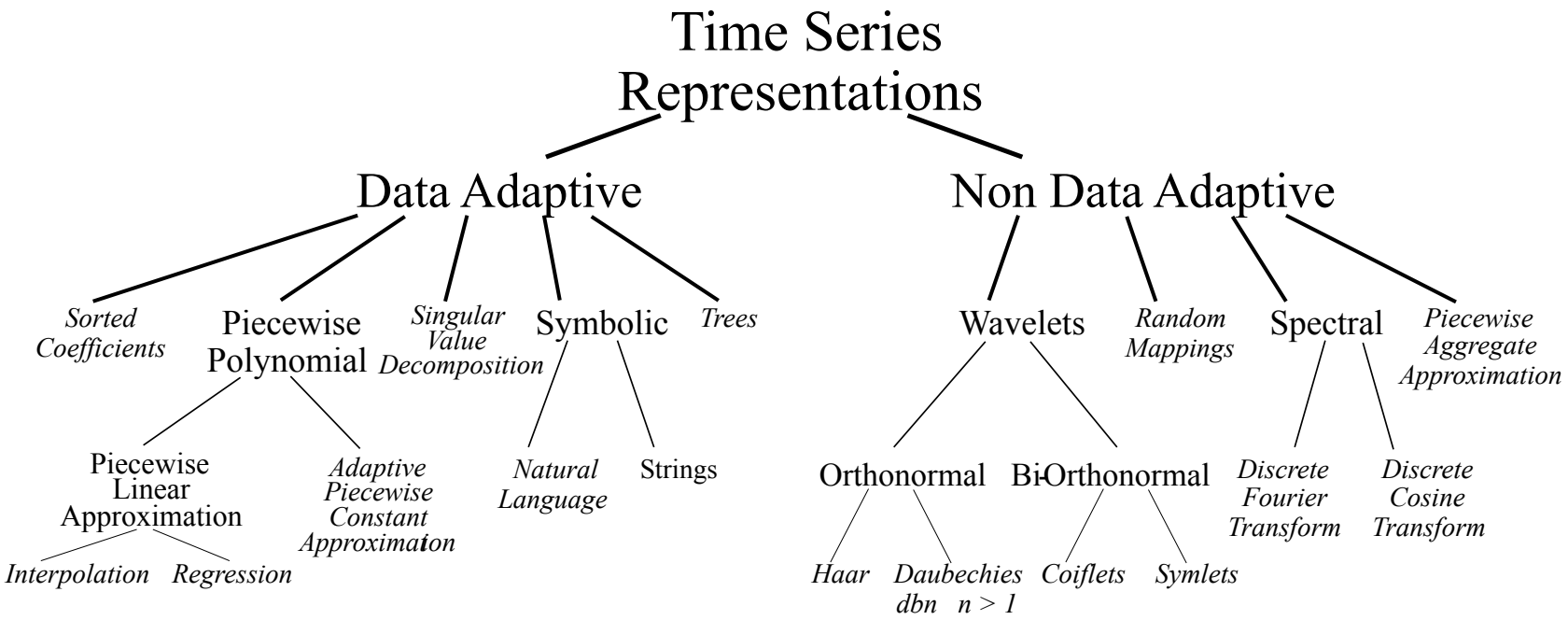
The Generic Data Mining Algorithm

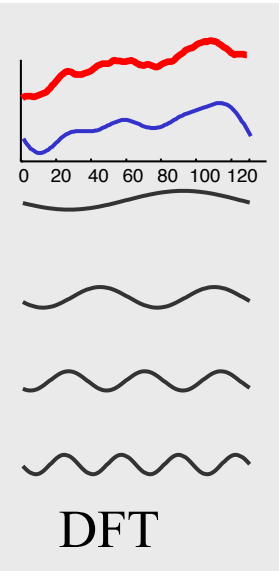
• Create an *approximation* of the data, which will fit in main memory, yet retains the essential features of interest

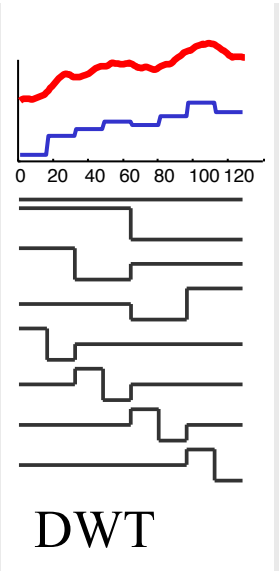
• Approximately solve the problem at hand in main memory

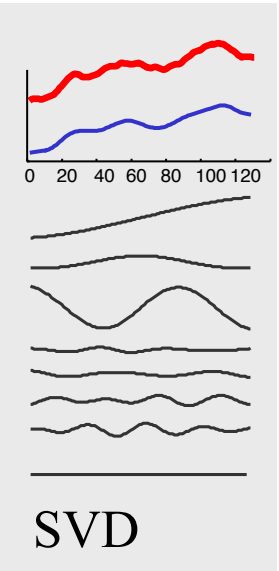
• Make (hopefully very few) accesses to the original data on disk to confirm the solution obtained in Step 2, or to modify the solution so it agrees with the solution we would have obtained on the original data

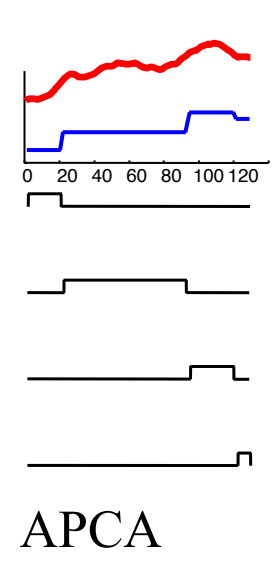


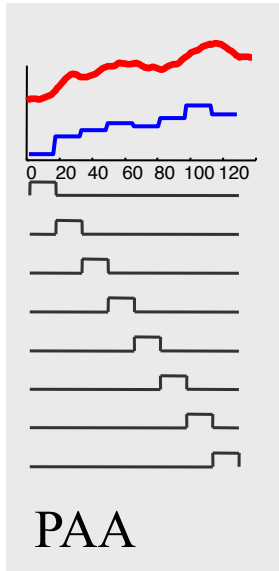

Time Series Representations
Data Adaptive
Non Data Adaptive
Sorted Coefficients
Piecewise Polynomial
Singular Value Decomposition
Symbolic
Trees
Wavelets
Random Mappings
Spectral
Piecewise Aggregate Approximation
Piecewise Linear Approximation
Adaptive Piecewise Constant Approximation
Natural Language
Strings
Orthonormal
Bi-Orthonormal
Discrete Fourier Transform
Discrete Cosine Transform
Interpolation
Regression
Haar
Daubechies dbn n > 1
Coiflets
Symlets

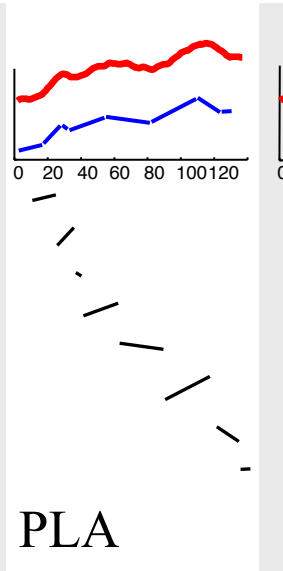

0 20 40 60 80 100 120
DFT

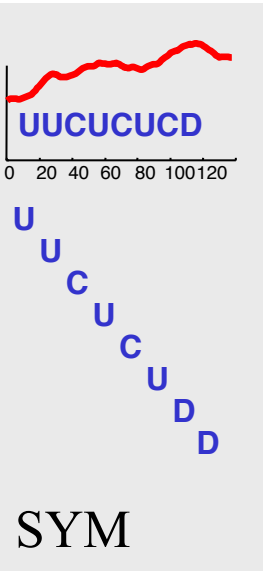

0 20 40 60 80 100 120
DWT


0 20 40 60 80 100 120
SVD


0 20 40 60 80 100 120
APCA


0 20 40 60 80 100 120
PAA


0 20 40 60 80 100 120
PLA

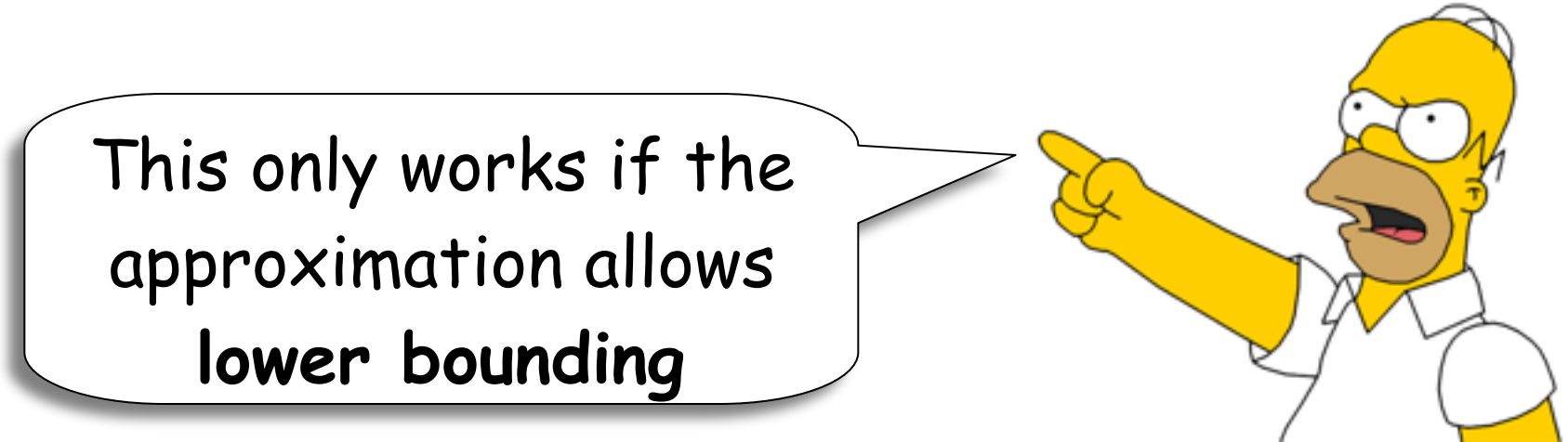

UUCUCUCD
0 20 40 60 80 100 120
U U C U C U D D
SYM

The Generic Data Mining Algorithm (revisited)

• Create an *approximation* of the data, which will fit in main memory, yet retains the essential features of interest

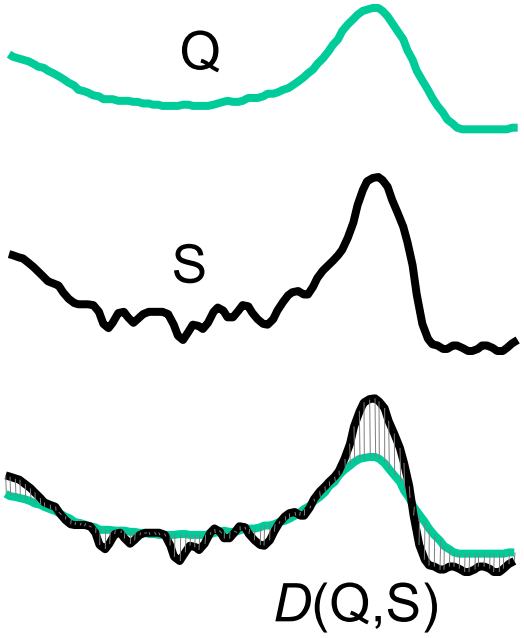
• Approximately solve the problem at hand in main memory

• Make (hopefully very few) accesses to the original data on disk to confirm the solution obtained in Step 2, or to modify the solution so it agrees with the solution we would have obtained on the original data



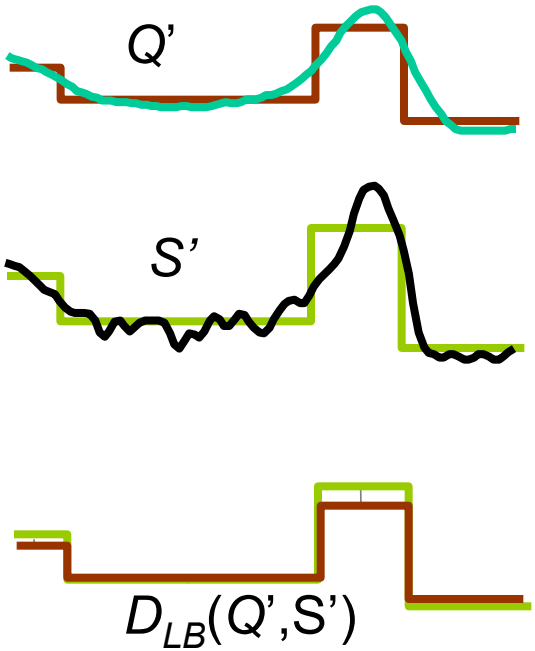
What is lower bounding?

Exact (Euclidean) distance D(Q,S)



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

Lower bounding distance $D_{LB}(Q,S)$

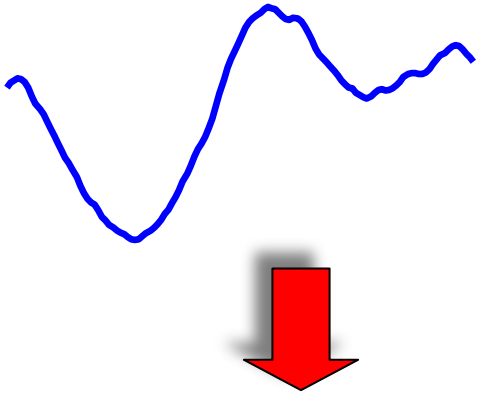


$$D_{LB}(Q',S') \equiv \sqrt{\sum_{i=1}^{M} (sr_i - sr_{i-1})(qv_i - sv_i)^2}$$

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

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

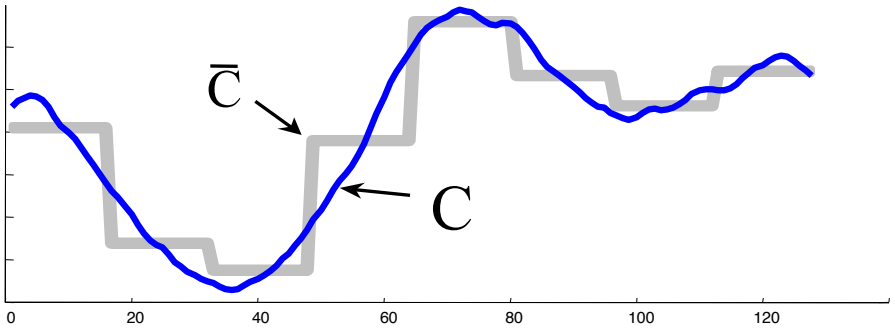
Symbolic **A**ggregate Appro**X**imation (SAX)



baabccbc

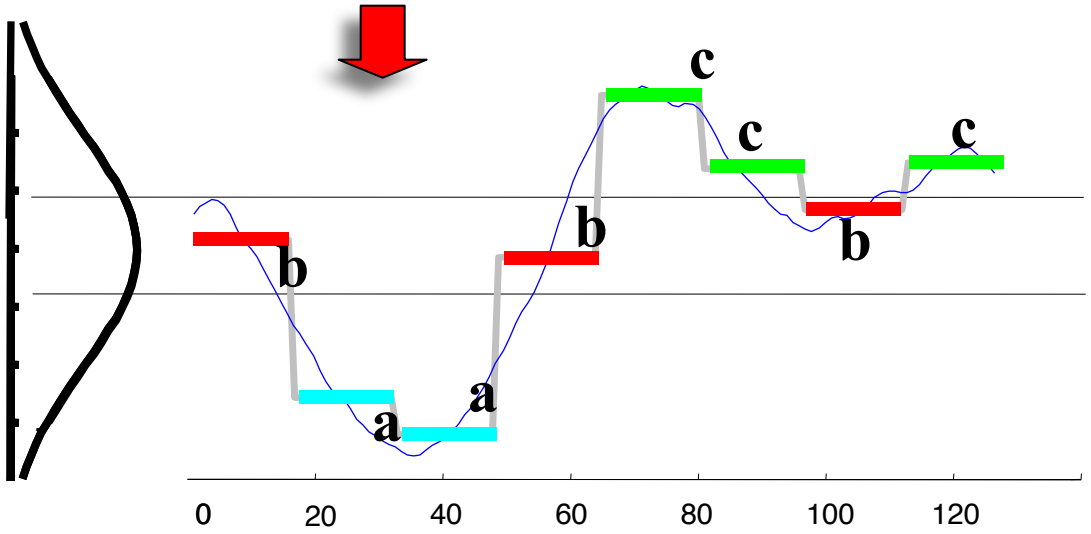
- Lower bounding of Euclidean distance
- Dimensionality Reduction
- Numerosity Reduction

How do we obtain SAX?

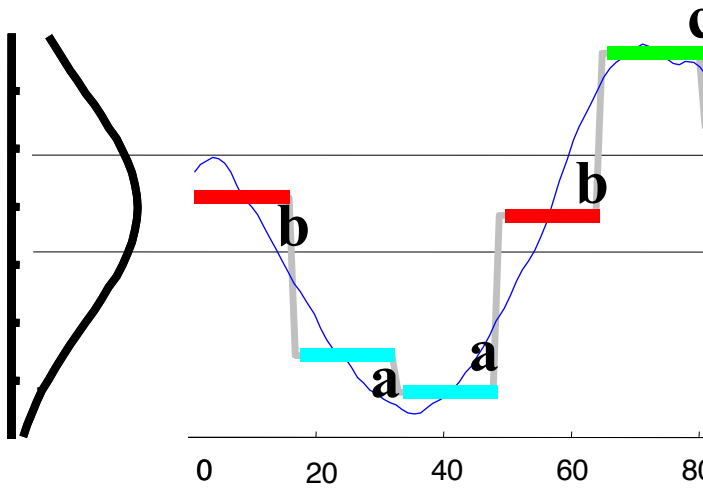


First convert the time series to PAA representation, then convert the PAA to symbols

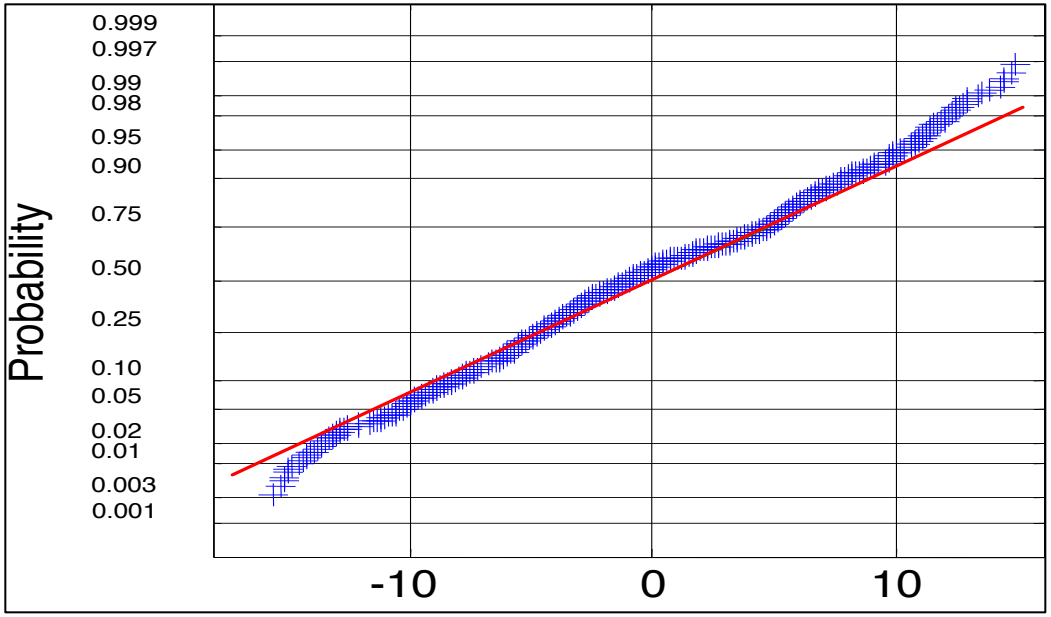
It take linear time



baabccbc

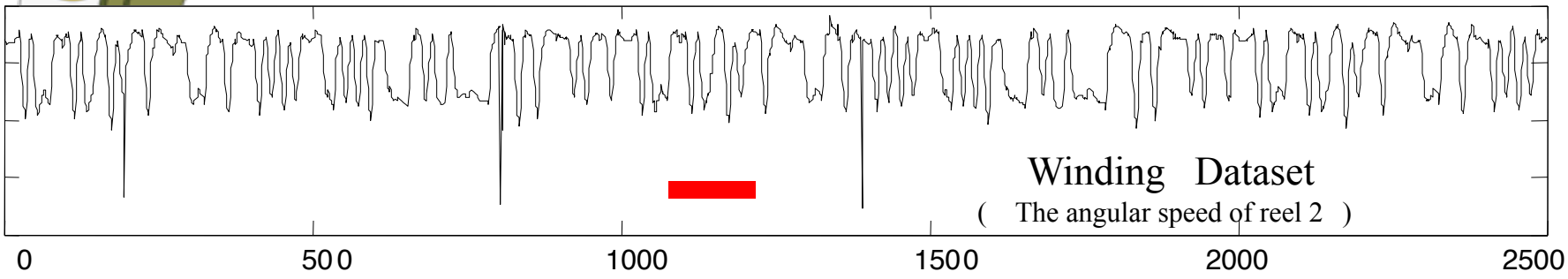


Time series subsequences tend to have a highly Gaussian distribution



A normal probability plot of the (cumulative) distribution of values from subsequences of length 128.

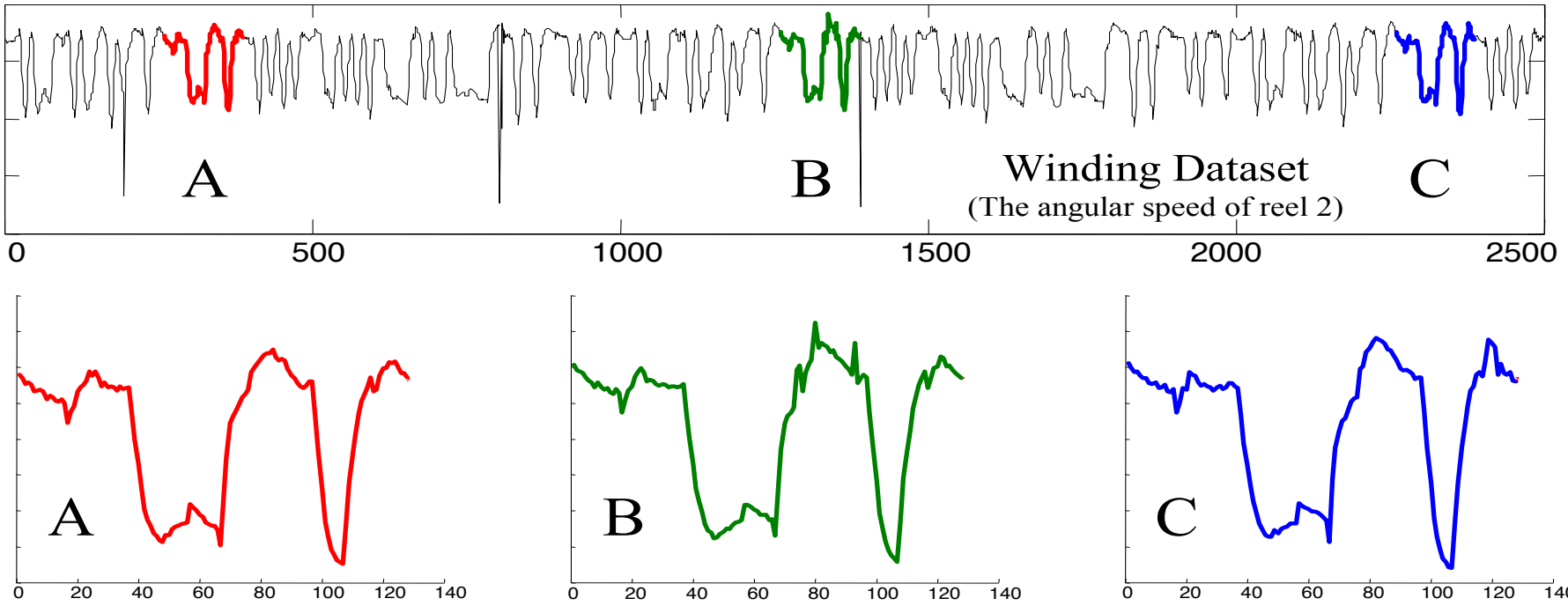
SAX allows Motif Discovery!



Informally, motifs are reoccurring patterns…

Motif Discovery

To find these 3 motifs would require about 6,250,000 calls to the Euclidean distance function.



OK, we can define motifs, but how do we find them?

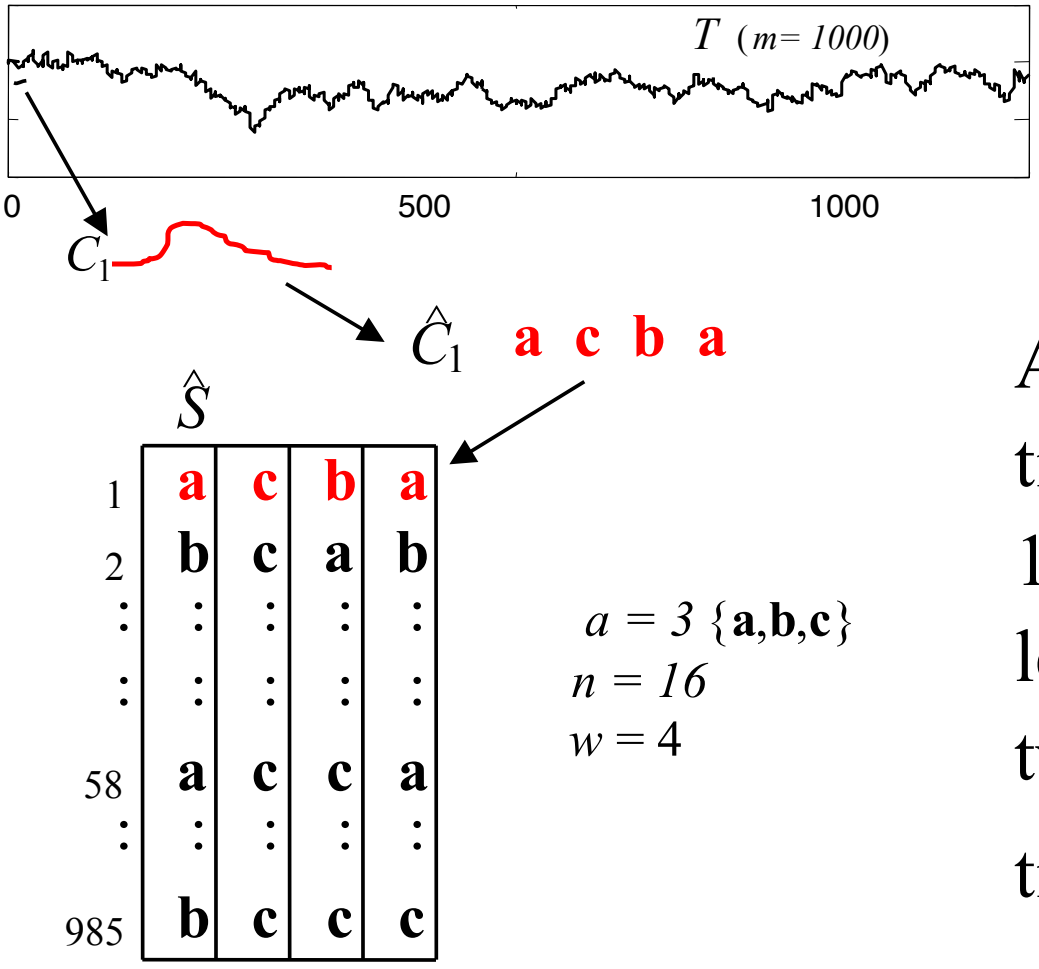
The obvious brute force search algorithm is just too slow…

Our 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.

*** J Buhler and M Tompa. *Finding motifs using random projections*. In RECOMB'01. 2001.**

A simple worked example of our motif discovery algorithm

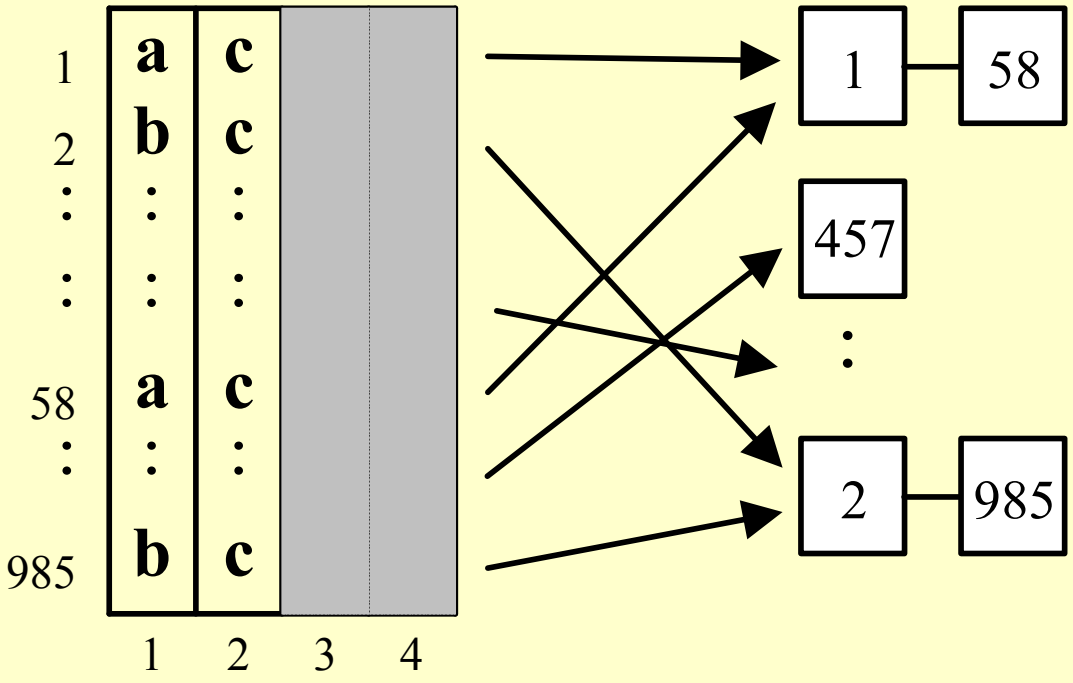
The next 4 slides



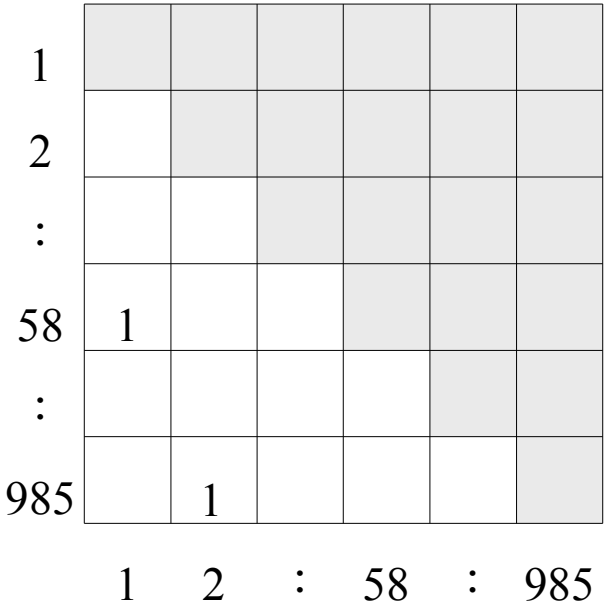
$a = 3$ {**a**,**b**,**c**}
$n = 16$
$w = 4$

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}.

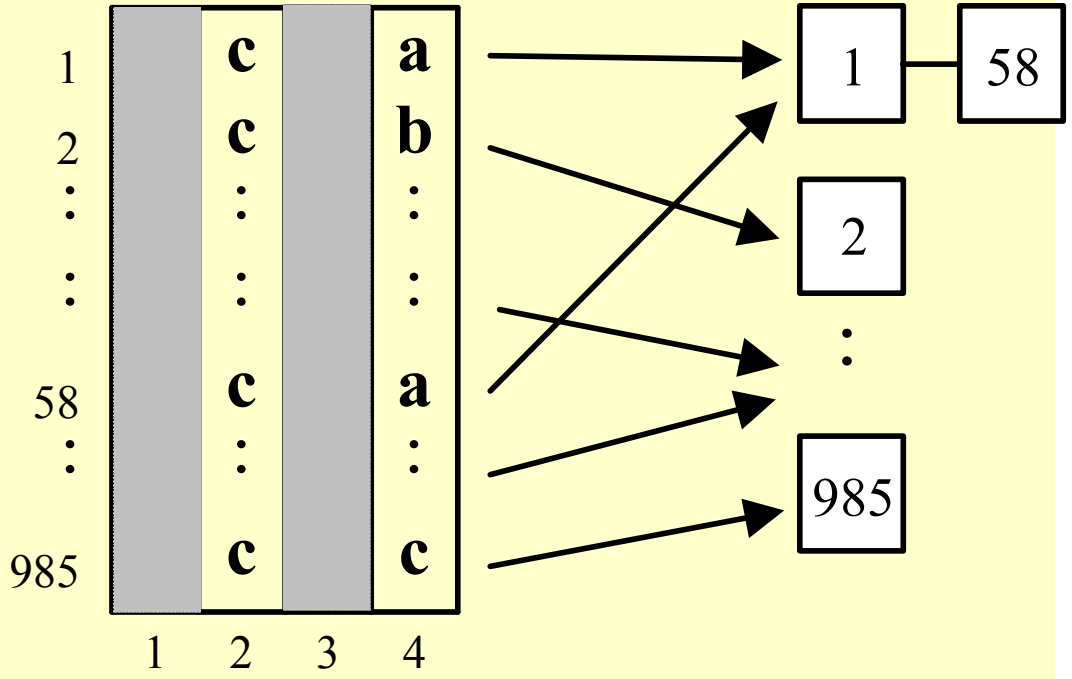
A mask {1,2} was randomly chosen, so the values in columns {1,2} were used to project matrix into buckets.



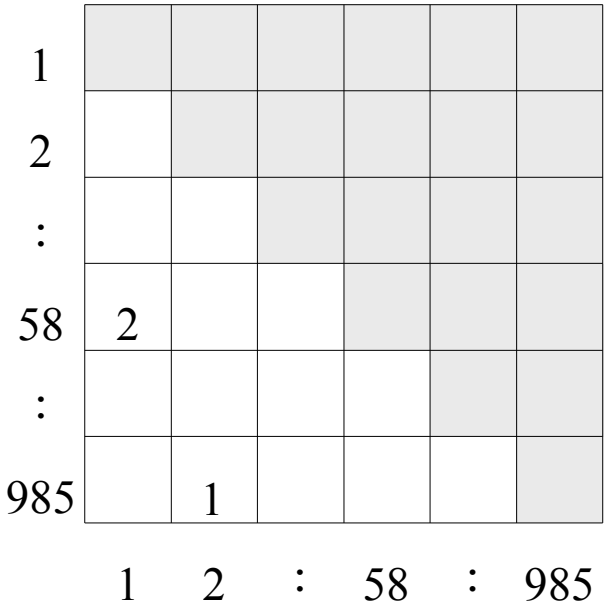
Collisions are recorded by incrementing the appropriate location in the collision matrix

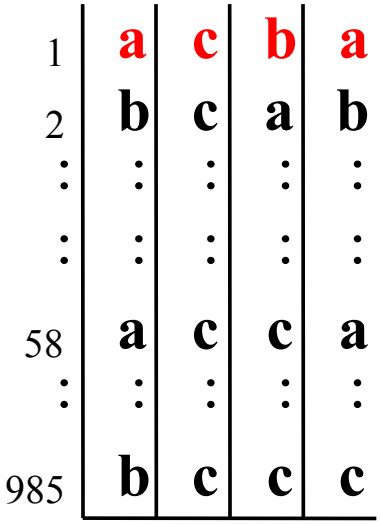


A mask {2,4} was randomly chosen, so the values in columns {2,4} were used to project matrix into buckets.



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





1						
2	2					
:	1	3				
58	**27**	2	1			
:	3	2	2	1		
985	0	1	2	1	3	
	1	2	:	58	:	985

Coming up…

Project Progress Report Due Today at 11:59PM

Panel of Experts (Thursday)

AR Models from Data and Human Computation

Panel of Experts (Next Tuesday)

Environmental Sensing: Objects and Simple Sensors