SAPIENZA
UNIVERSITÀ DI ROMA

Giovanni Stilo
stilo@di.uniroma1.it

SAX!
A Symbolic Representations of Time Series

What are Time Series?

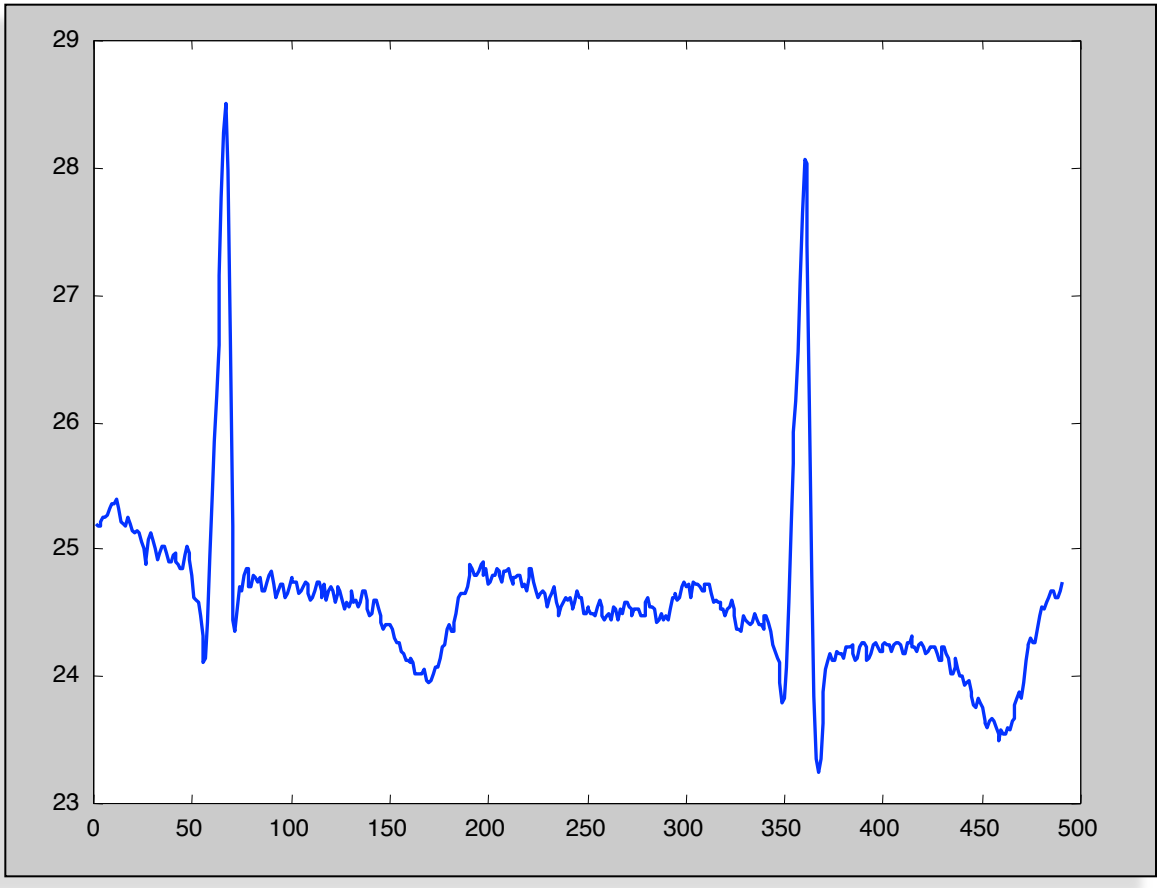
25.1750
25.2250
25.2500
25.2500
25.2750
25.3250
25.3500
25.3500
25.4000
25.4000
25.3250
25.2250
25.2000
25.1750

..

..

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.



Time Series are Ubiquitous

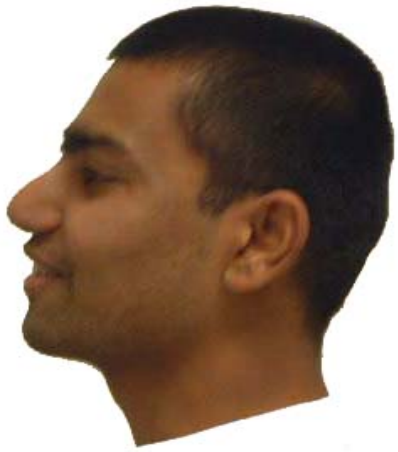
People measure things...

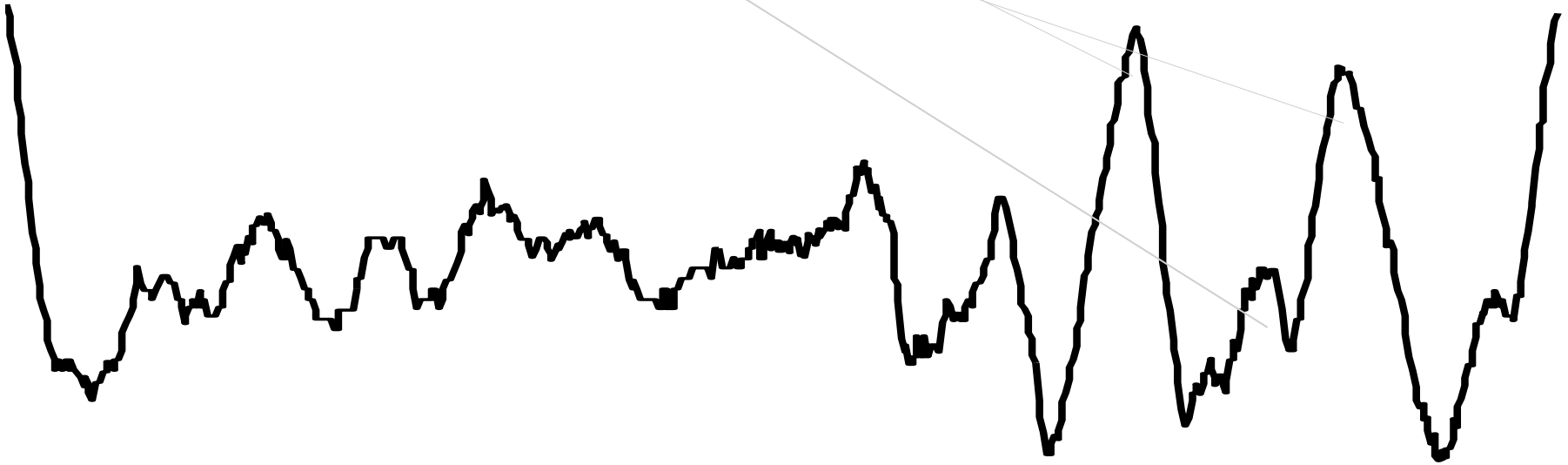
- *Schwarzeneggers popularity rating.*
- *Their blood pressure.*
- *The annual rainfall in New Zealand.*
- *The value of their Yahoo stock.*
- *The number of web hits per second.*

... and things change over time.

Thus time series occur in virtually every medical, scientific and businesses domain.

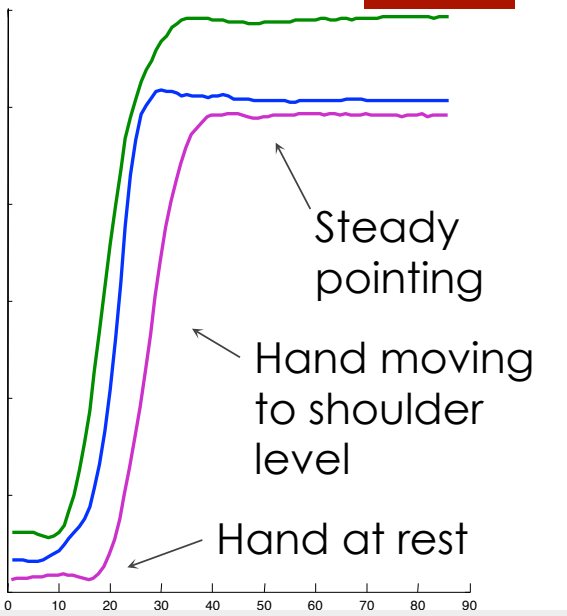
Image data, may best be thought of as time series…



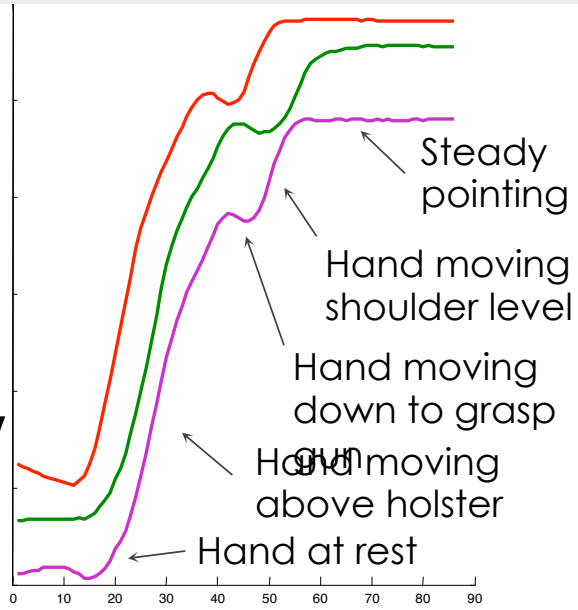


Video data, may best be thought of as time series…

Point

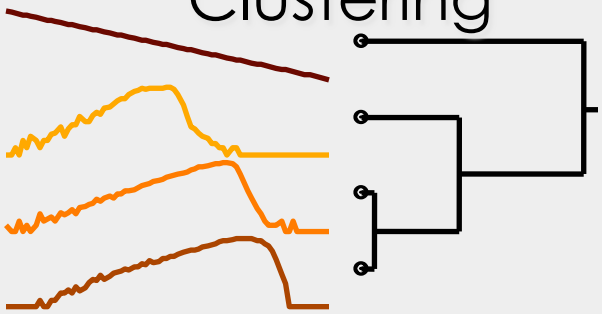


Gun-Draw

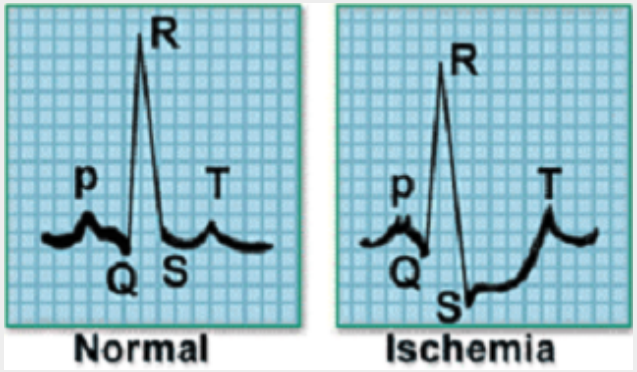


What do we want to do with the time series data?

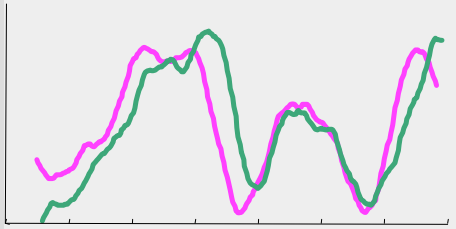
Clustering



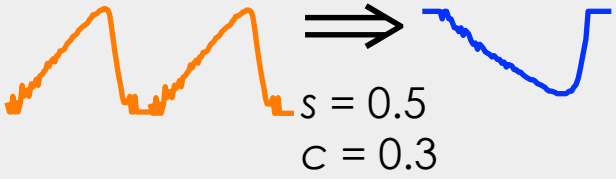
Classification



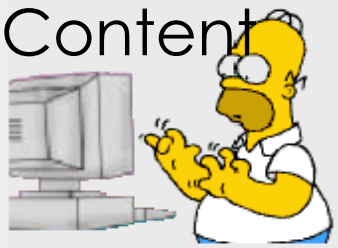
Motif Discovery

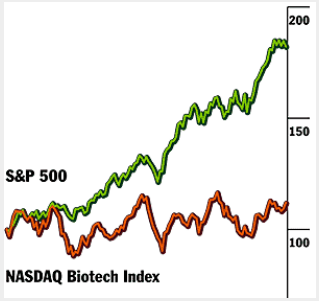


Rule Discovery

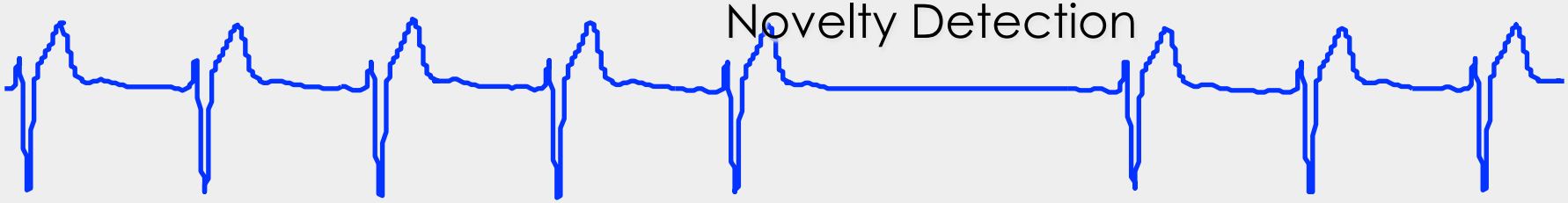


Query by Content

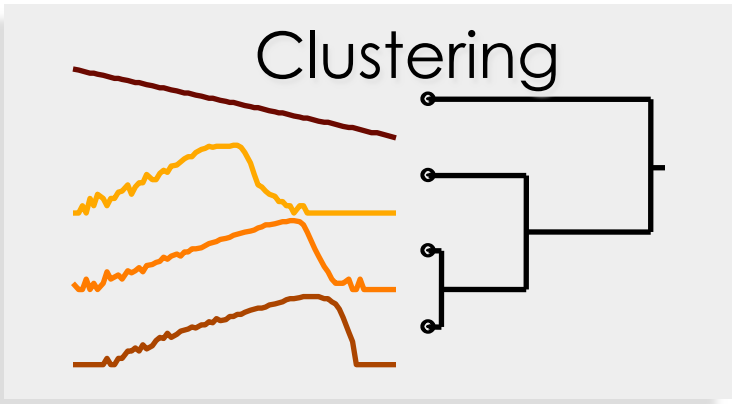


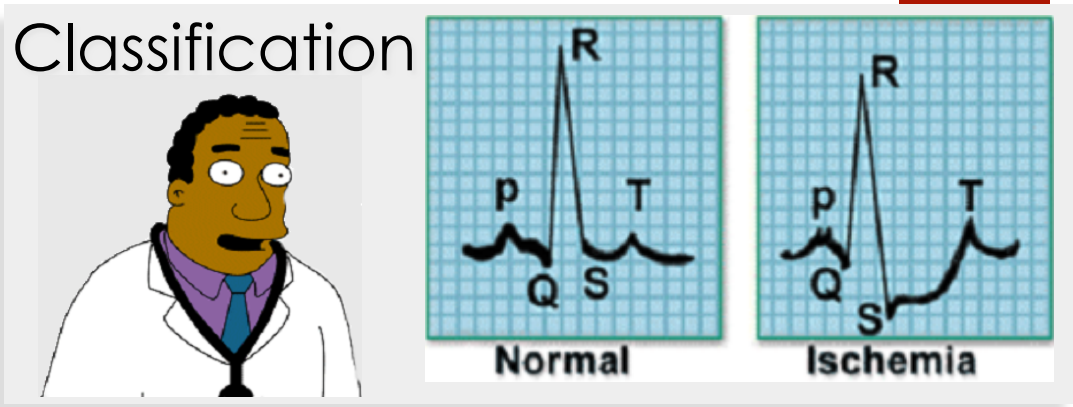


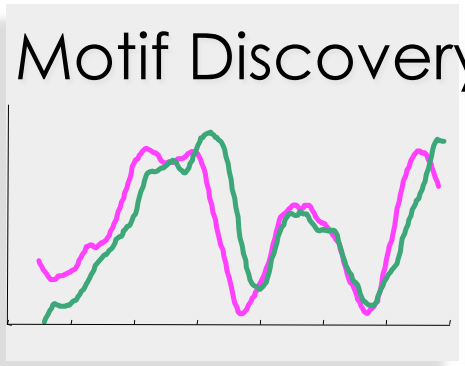
Novelty Detection

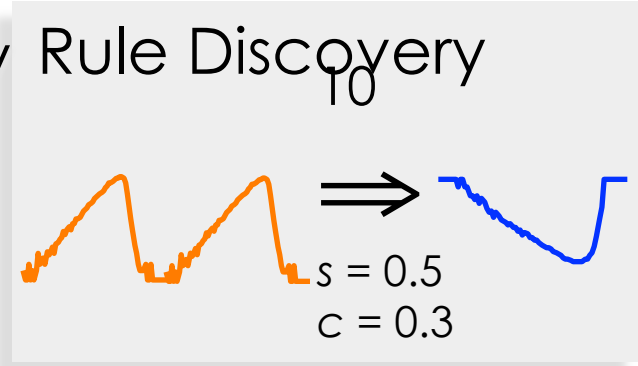


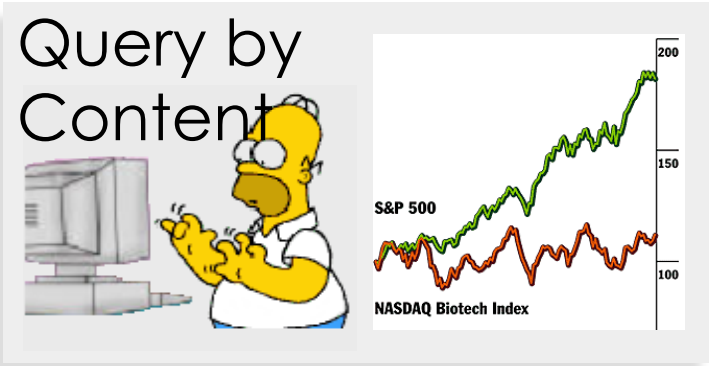
All these problems require **similarity** matching

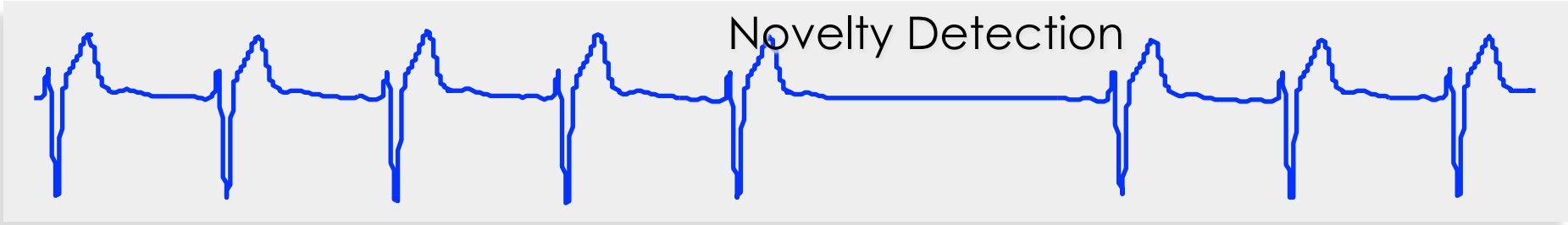












Euclidean Distance Metric

Given two time series

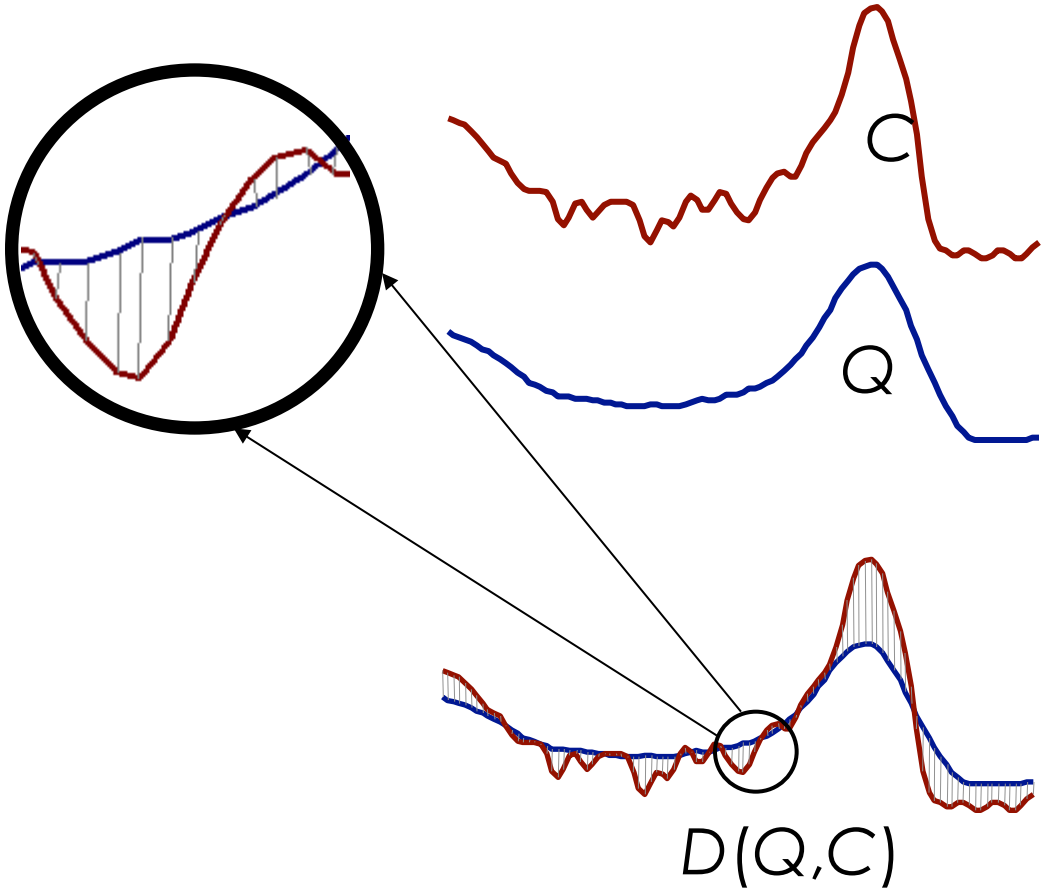
$Q = q_1 \ldots q_n$

and

$C = c_1 \ldots c_n$

their Euclidean distance is defined as:

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

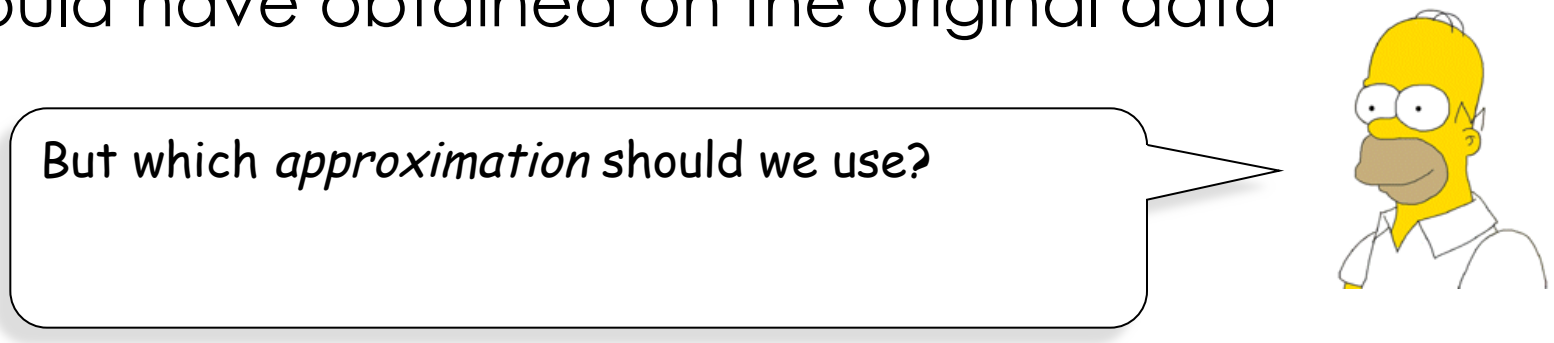


The Generic Data Mining Algorithm

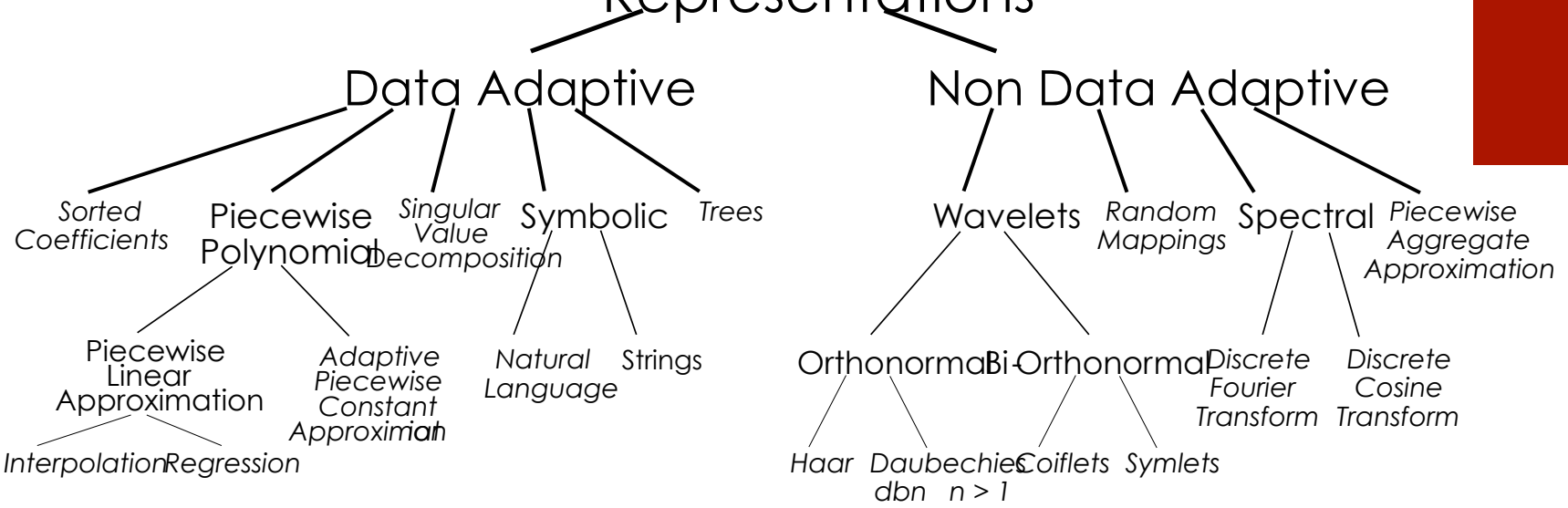
- Create an *approximation* of the data, which will fit in main memory, yet retains the essential features of interest

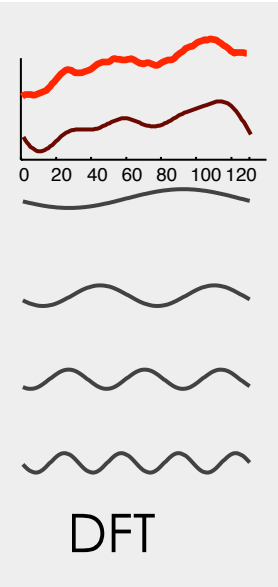
- Approximately solve the problem at hand in main memory

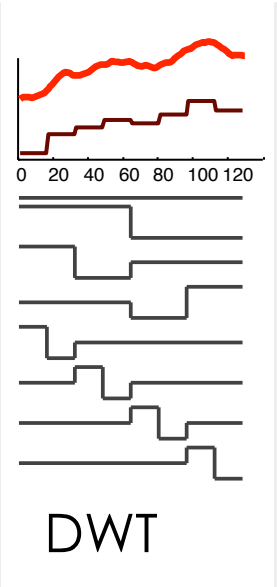
- Make (hopefully very few) accesses to the original data on disk to confirm the solution obtained in Step 2, or to modify the solution so it agrees with the solution we would have obtained on the original data

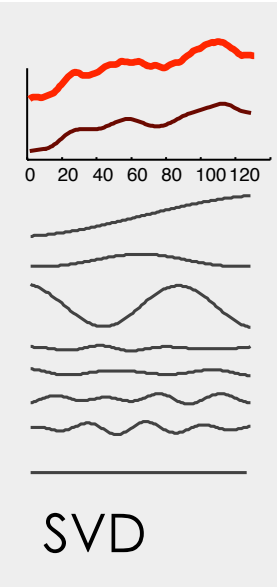


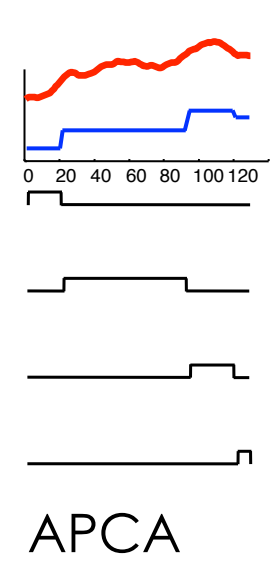
Time Series Representations

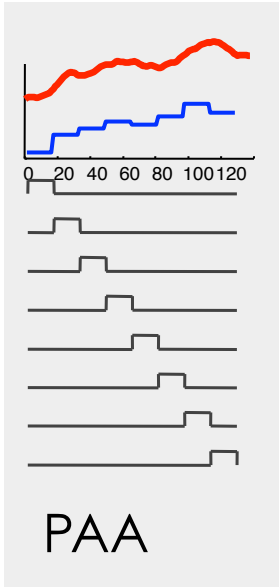


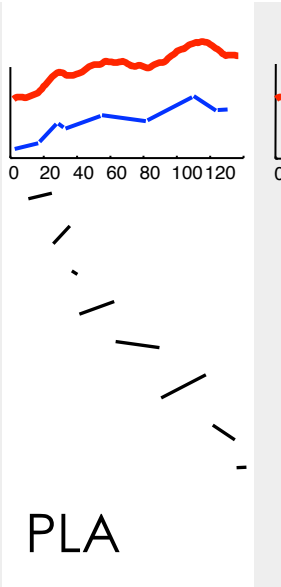


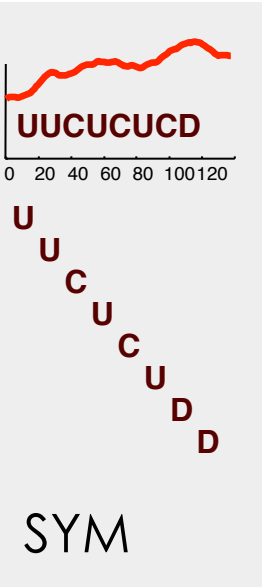










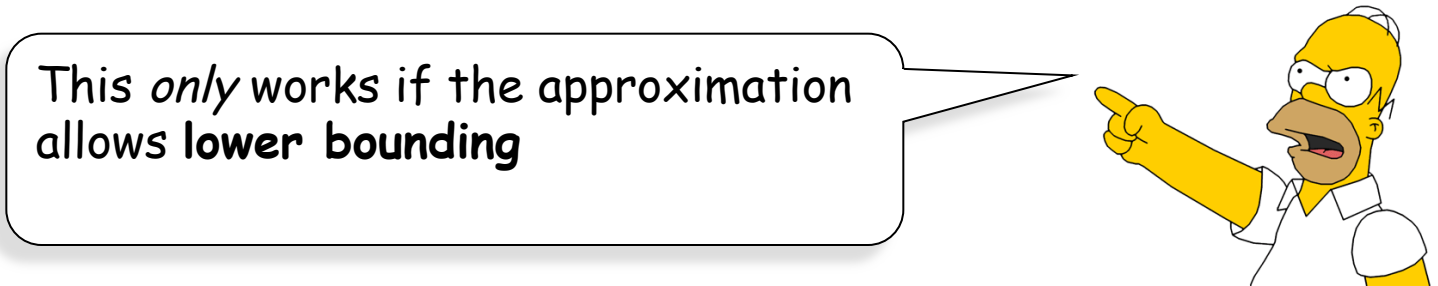


The Generic Data Mining Algorithm

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

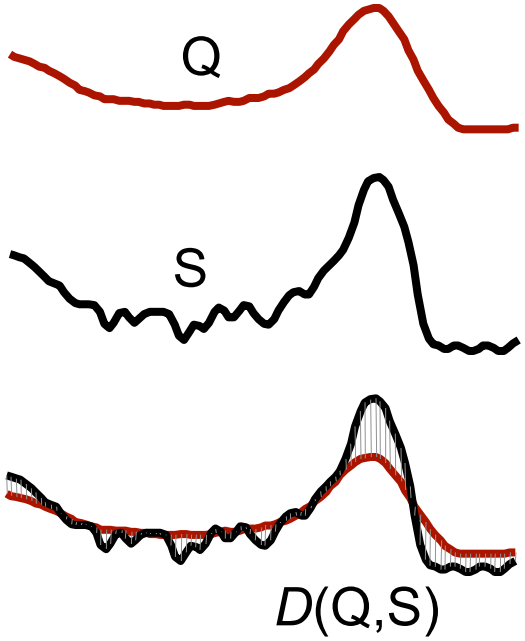
- Approximately solve the problem at hand in main memory

- 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

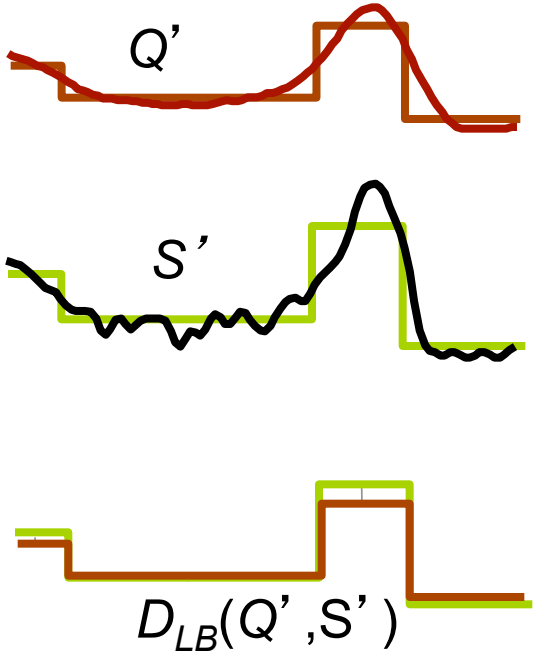


What is lower bounding?

Exact (Euclidean) distance D(Q,S)



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

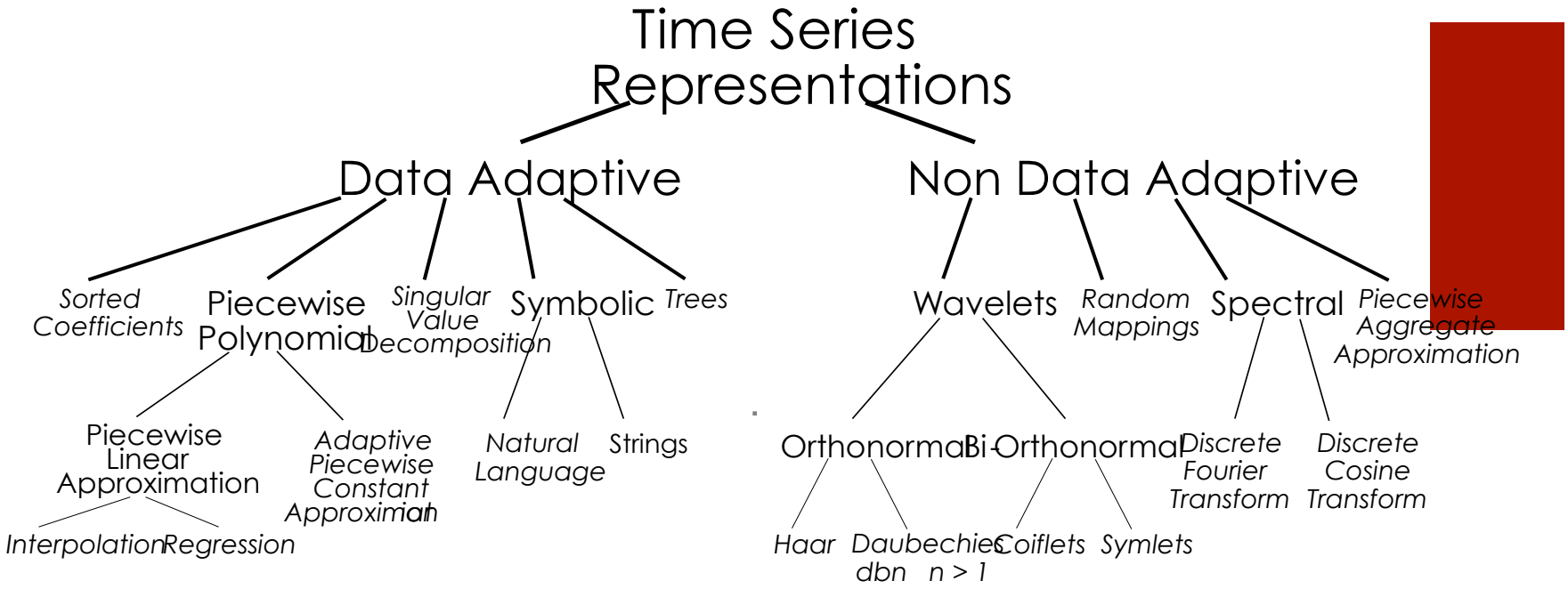


$$D(Q,S) \equiv \sqrt{\sum_{i=1}^{n} (q_i - s_i)^2}$$

$$D_{LB}(Q',S') \equiv \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (\bar{q}_i - \bar{c}_i)^2}$$

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

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



We can live without "*trees*", "*random mappings*" and "*natural language*", but it would be nice if we could lower bound *strings (symbolic or discrete approximations)*…

A lower bounding symbolic approach would allow data miners to…

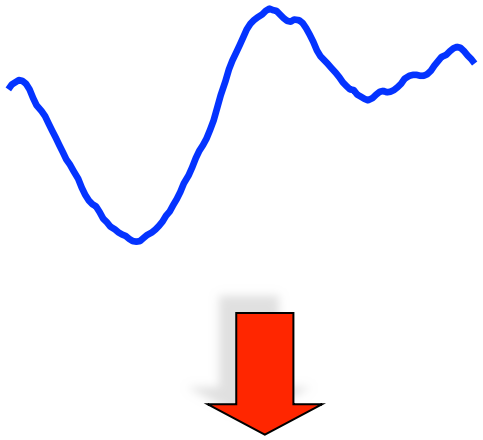
- Use suffix trees, hashing, markov models etc
- Use text processing and bioinformatic algorithms

The first symbolic representation of time series, that allows…

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

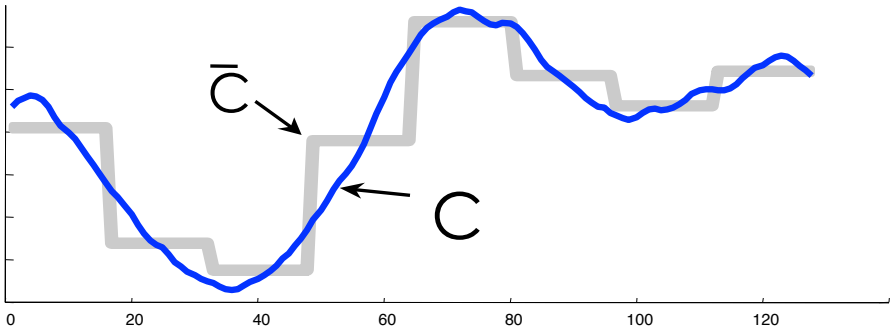
This representation SAX

Symbolic **A**ggregate Appro**X**imation



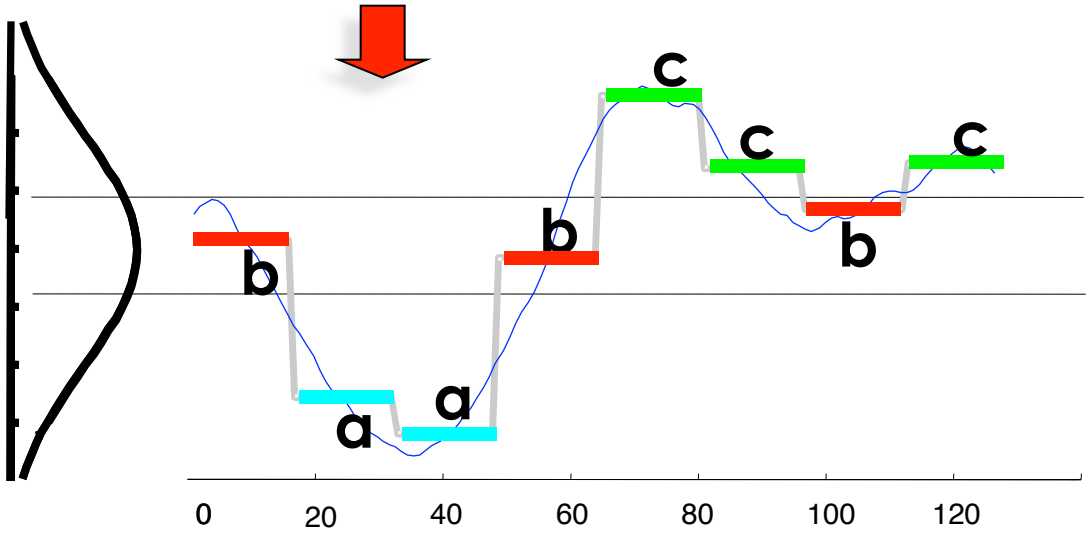
baabccbc

ow do we obtain SAX?

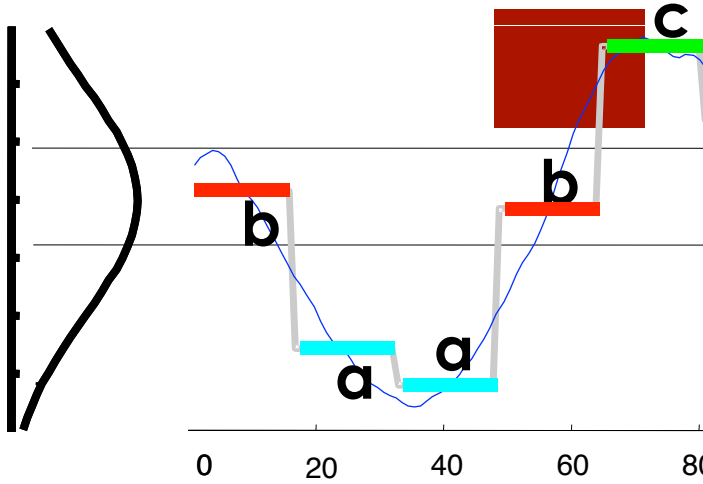


First convert the time series to PAA representation, then convert the PAA to symbols

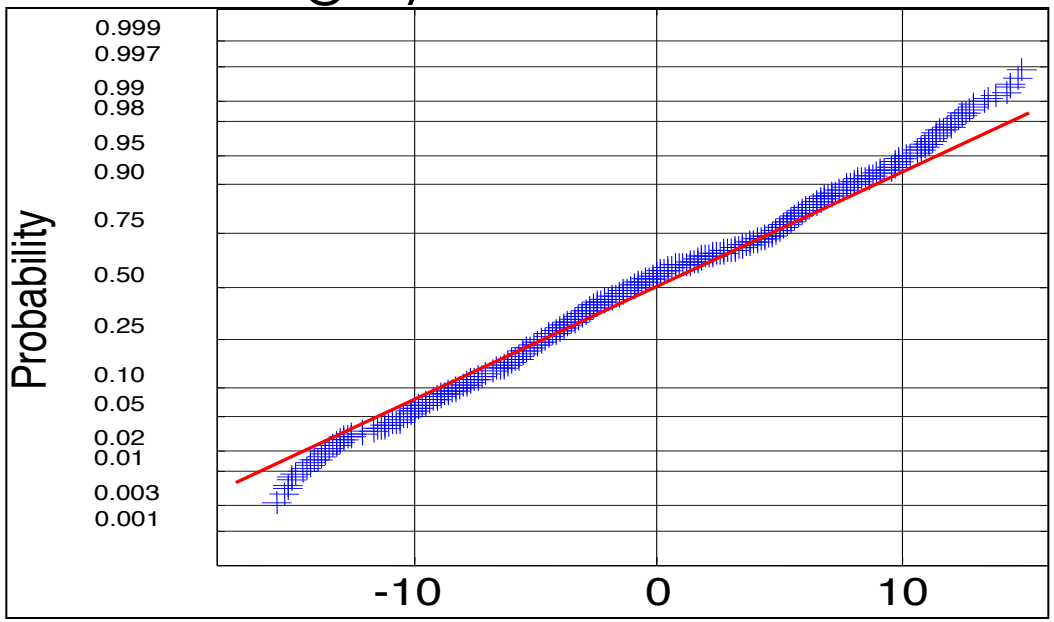
It take linear time



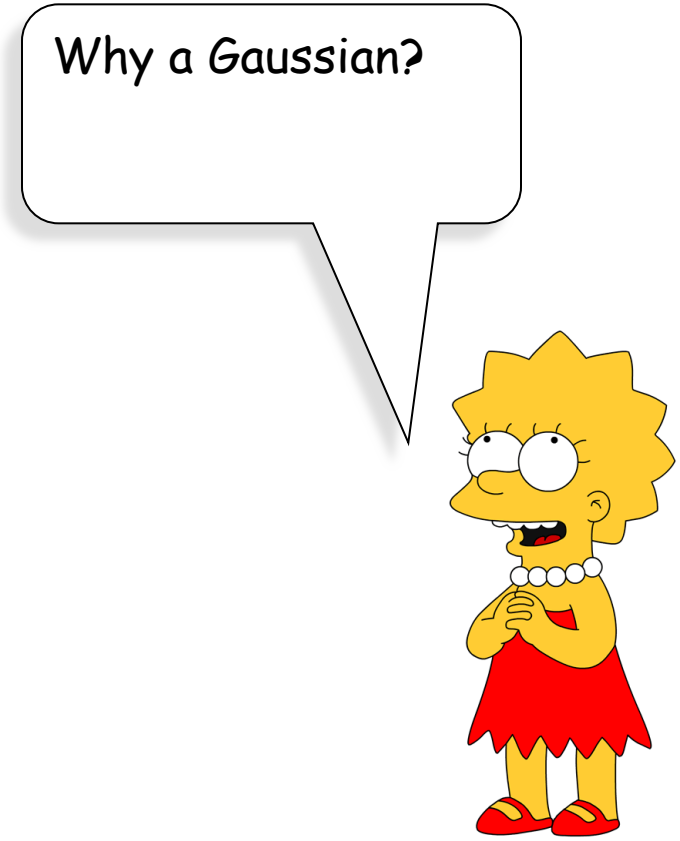
baabccbc



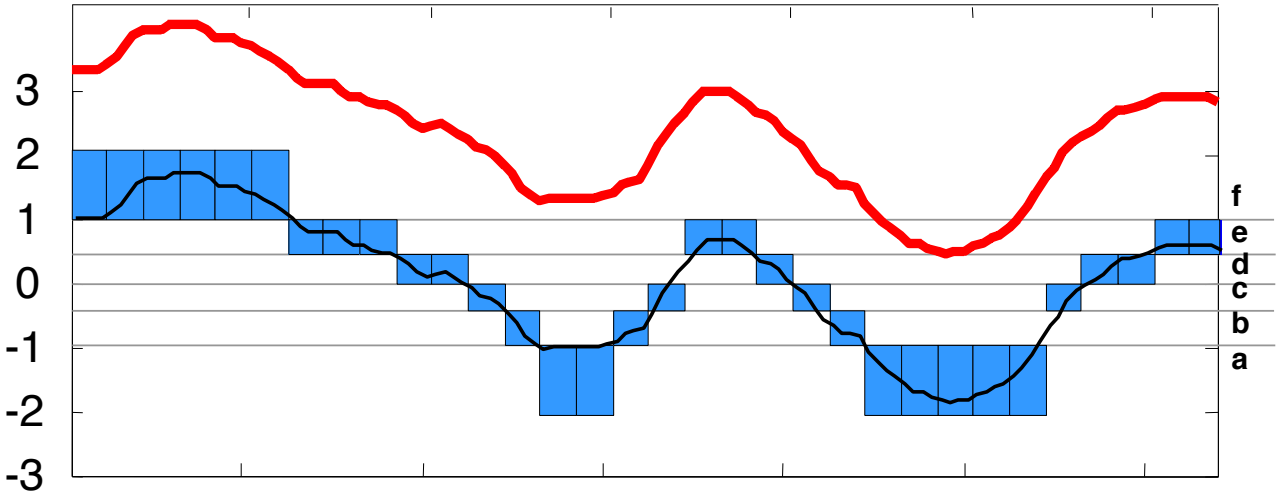
Time series subsequences tend to have a highly Gaussian distribution

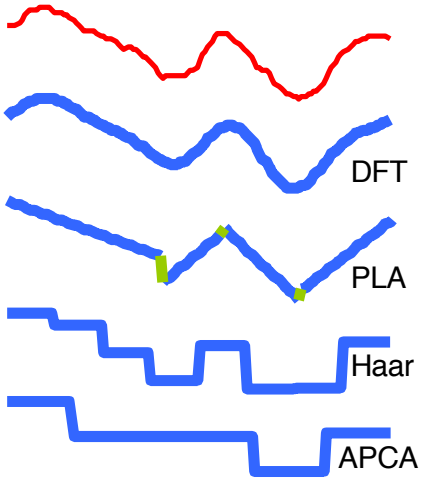


A normal probability plot of the (cumulative) distribution of values from subsequences of length 128.



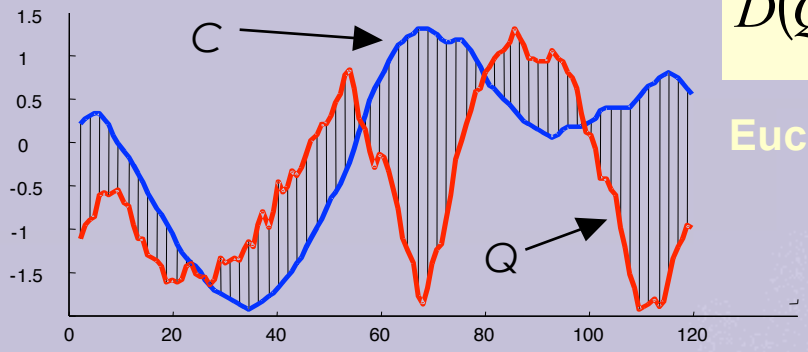
Visual Comparison





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

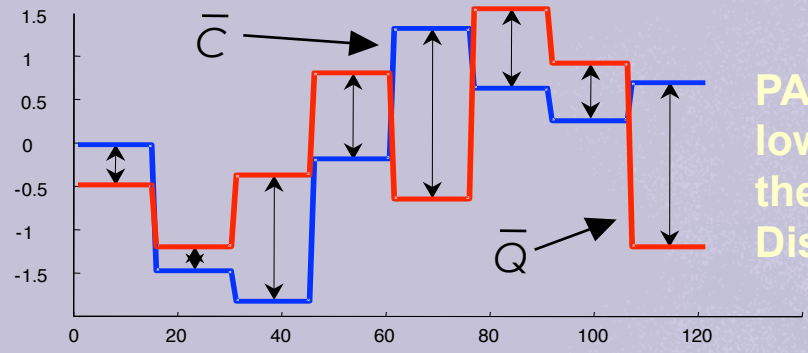
- We can use more symbols to represent the time series since each symbol requires fewer bits than real-numbers (float, double)



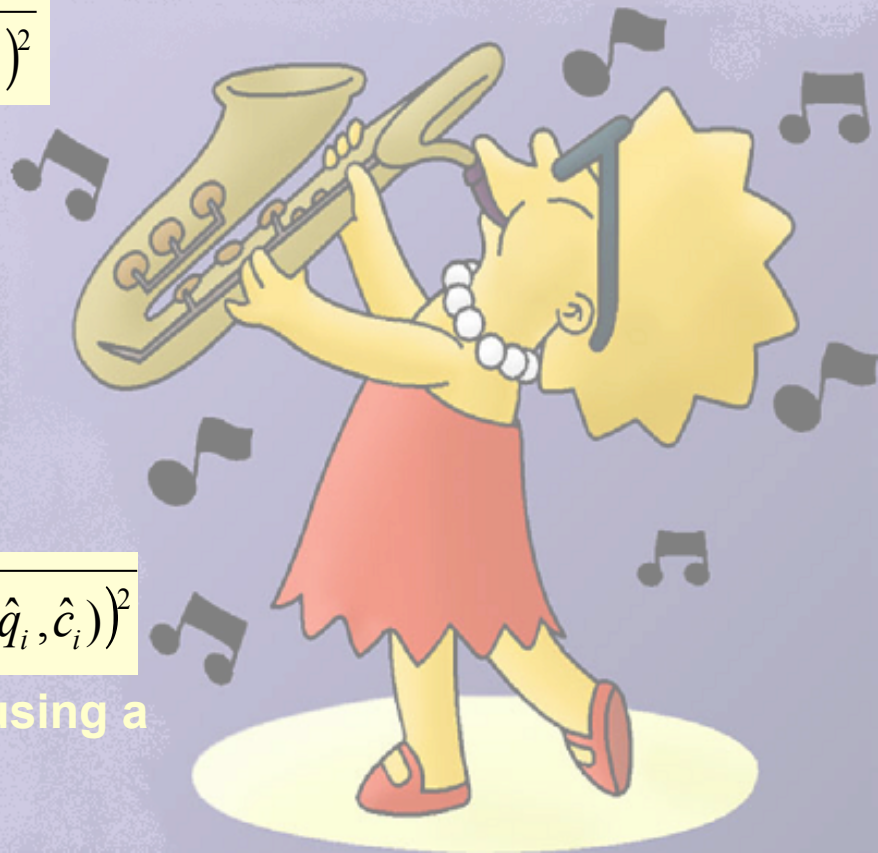
$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2}$$

Euclidean Distance

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



PAA distance lower-bounds the Euclidean Distance



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

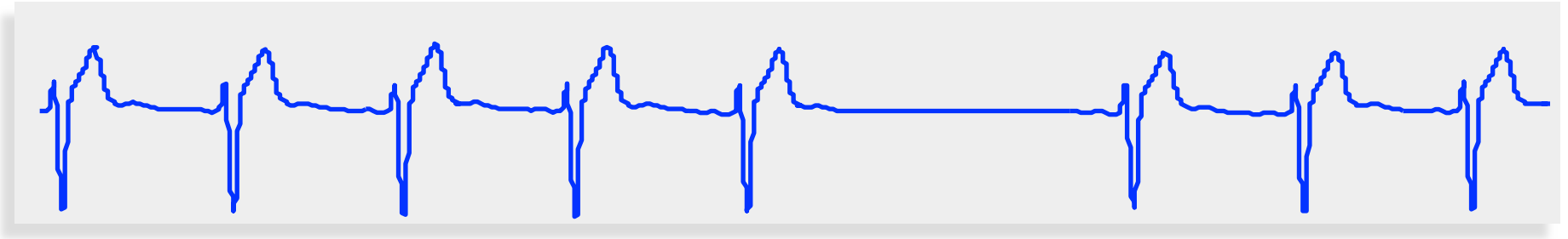
$\hat{C}$ = **baabccb**

$\hat{Q}$ = **babcacca**

dist() can be implemented using a table lookup.

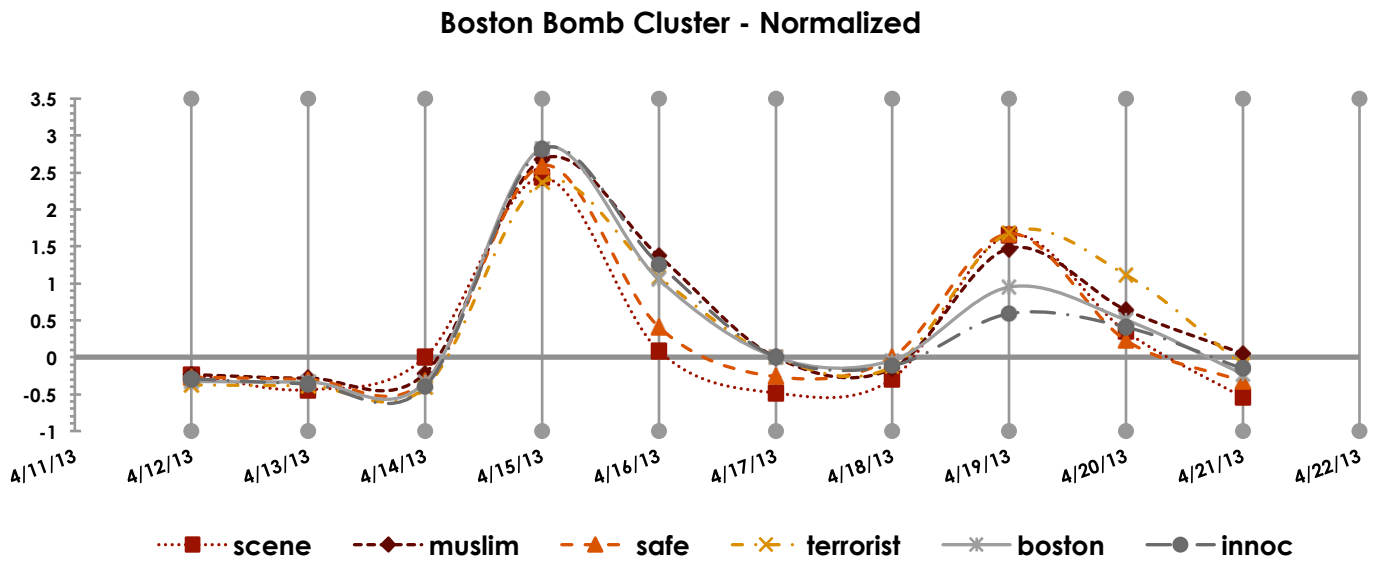
Novelty Detection

- Fault detection
- Interestingness detection
- Anomaly detection
- Surprisingness detection



A temporal notion of "meaning"

- Words with similar temporal behavior are similar
- "similar" = same time-frame, similar shape
- "similar" = **synonym** (#papafrancesco,#newpope) **OR contextually related** (Boston, bomb, marathon)



Temporal Data Mining

- "discovering temporal patterns in text information collected over time" (Mei and Zhai, 2005).
- When applied to large and lengthy micro-blog stream main problem is **complexity**.
  - Need efficient way of representing continuous signals
  - Need methods to prune non-relevant signals/identify "relevant" patterns
  - Need algorithms to efficiently detect similarity among signals
- Evaluation is also an issue: millions of often "obscure" patterns, lack of golden datasets and benchmarks

SAX*: temporal clustering based on Symbolic Aggregate Approximation

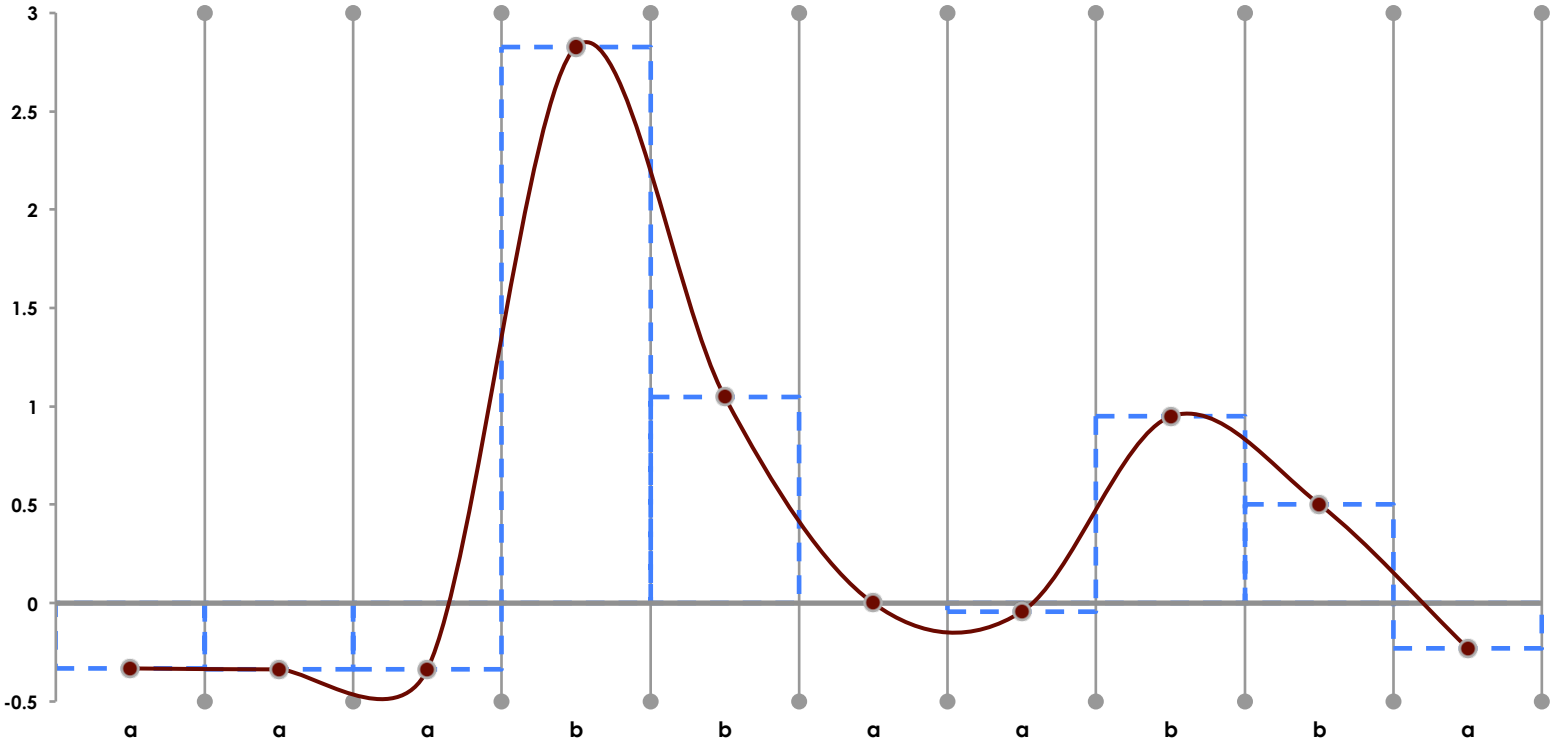
- 3 steps:
  1. Convert signals into sequences of symbols using Symbolic Aggregate Approximation
  2. Learn patterns of collective attention (regular expression)
  3. Cluster "relevant" strings in sliding windows

1. Symbolic Aggregate Approximation

- **Parameters**: W (dimension of temporal window) Δ (discretization step) Σ (alphabet of symbols)
- Signal is first Z-normalized
- In each (sliding) window W, signal is partitioned **vertically** in W/Δ slices, and the average value *is* computed in each slot Δj
- Signal is partitioned **horizontally** in | Σ | slices of equal area, let β j be the breackpoints
- A symbol is associated to every Δj according to:

$$si=j,\ j \in \Sigma,\ iff\ \beta j-1 < si < \beta j$$

Example (two symbols, breackpoint is 0)



W=10, Δ=1, Σ= a,b
string is aaabbaabba

2.Patterns of collective attention

- Apply SAX to manually selected (about 30) words related to known event from Wikipedia Events descriptions (Arab Spring, Olympics, tsunami, Occupy wall Street..)
- Generate compatible regular expressions using RPNI algorithm (Oncina and Garcia 1992)
- The following regex is learned(one-two peaks/plateaux):

a+b?bb?a+?a+b?bba*?

- Turns out to be compatible with shapes graphically shown in previous works on clustering patterns of collective attention (Lehmann et al. 2012; Xie et al. 2013; Weng et Lee 2011; Yang and Leskovec 2011)

3. Pattern clustering

- Bottom-up hierarchical clustering algorithm with *complete linkage* (Jain 2010)
- stop hierarchical bottom-up clustering aggregation for a cluster when:

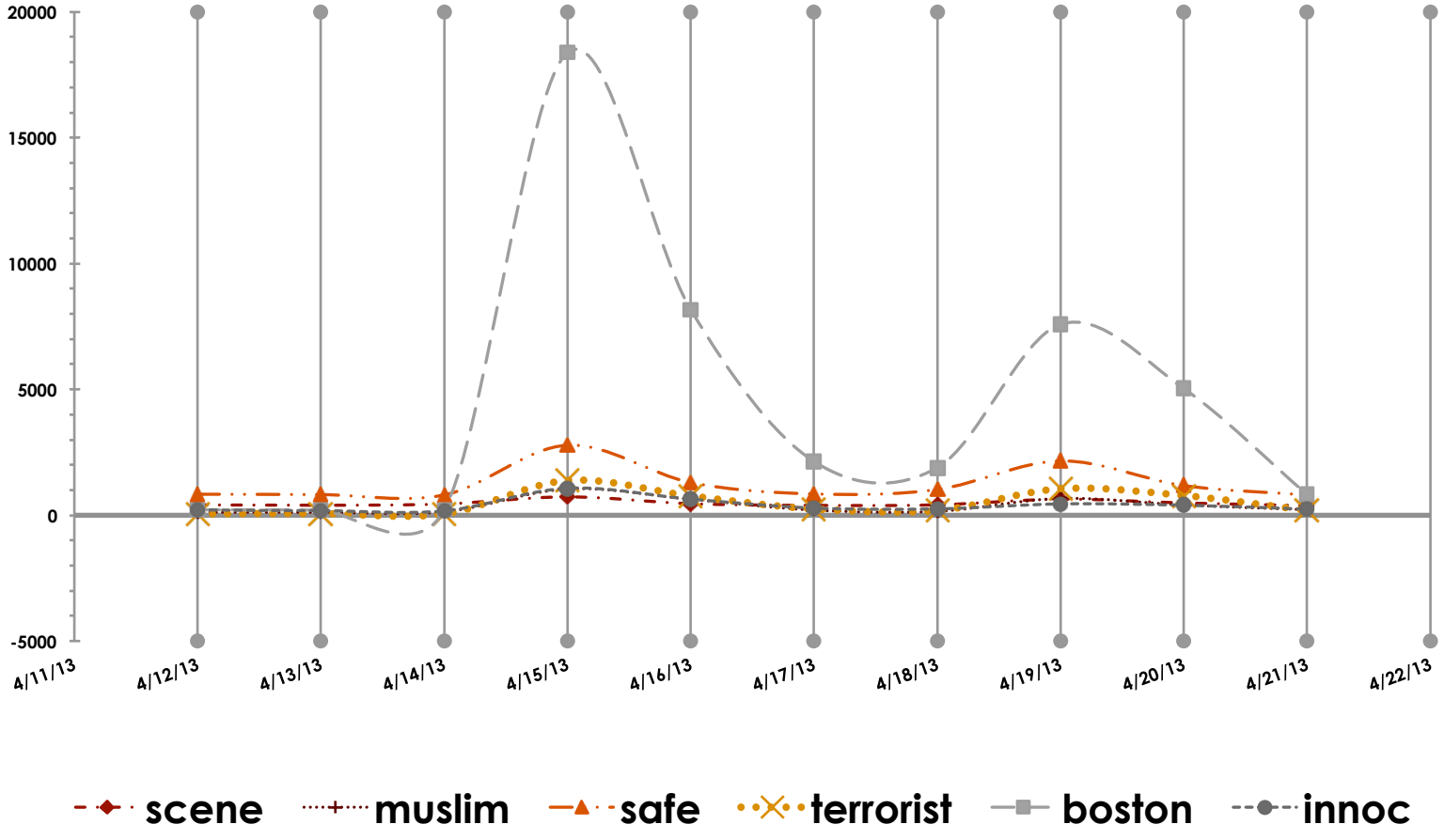
$$SD(d(centroid,tk)) < \delta$$

- Clusters smaller than *f* elements are purged
- To summarize, parameters are: W, Δ, Σ, δ, f
- Clusters are created in sliding windows of lenght W; "Δ-clusters" are subsequently generated (clusters active in slot Δ)

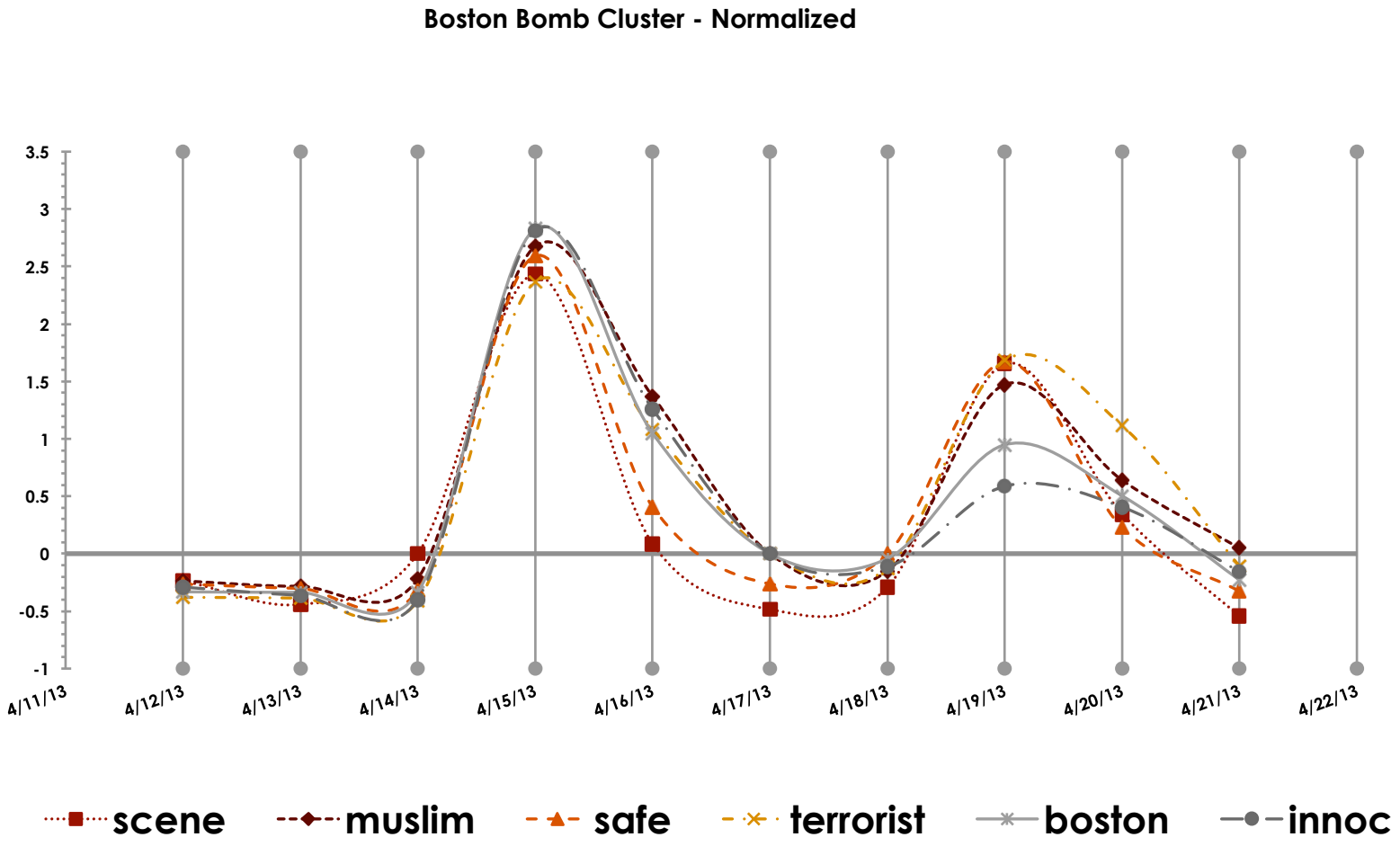
Experiments

- 1 year 1% Twitter stream (2012-13) ~7TB of Data.
- 2 types of experiments: **event detection** (in English) and (multilingual) **hashtag sense clustering**
- Parameters setting (especially W, Δ, Σ) depend on "density" and locality of stream. With 1% world-wide stream only "big events" can be detected.
- Best results with W=10 days, Δ=1 day, Σ=a,b
- For each experiment (events, hashtags), SAX* clustering is performed 365 times (one for each sliding window Wi).

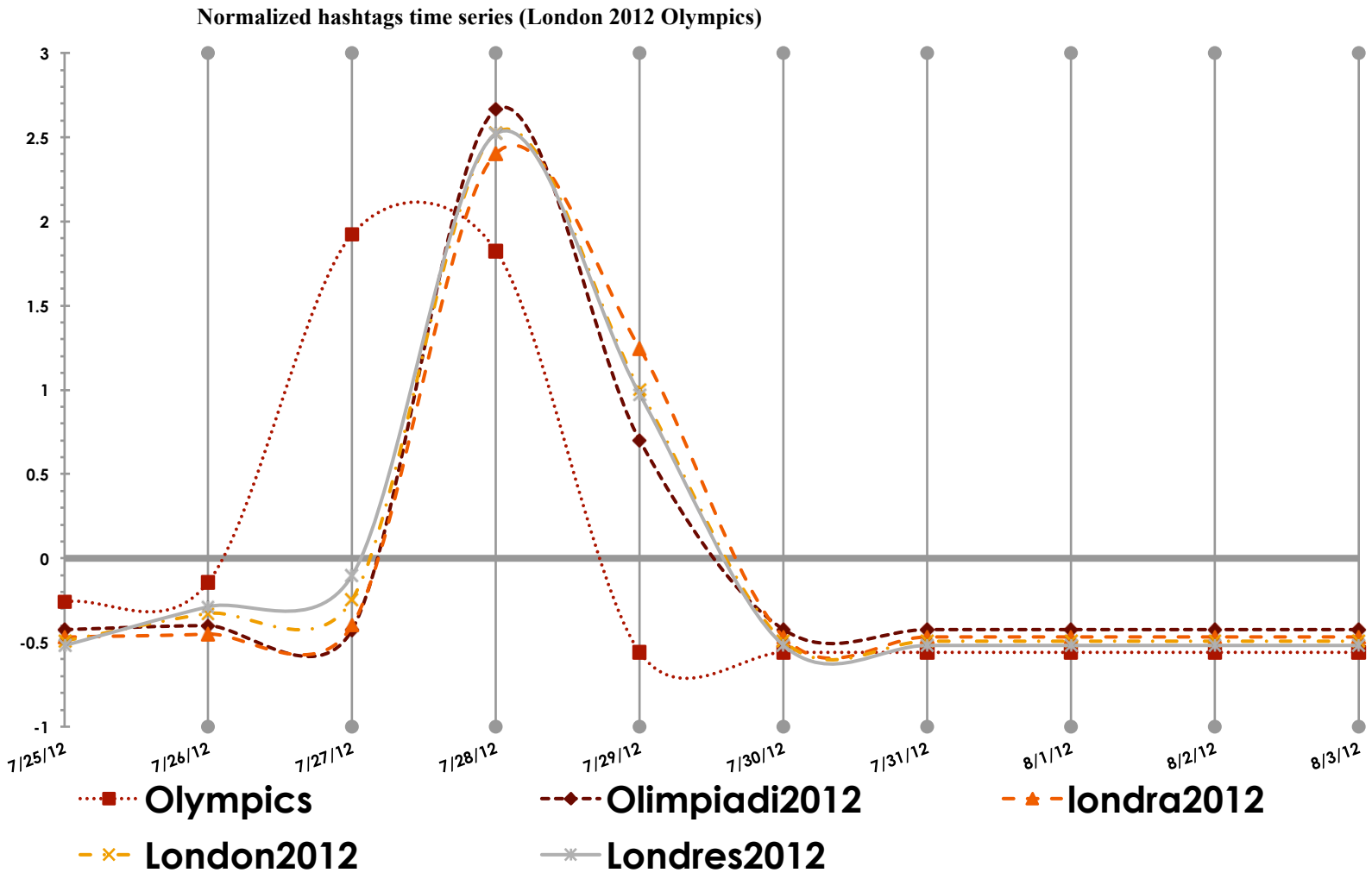
Example: "Bomb during Boston marathon, non normalized



Example: “Bomb during Boston marathon, normalized



Example: London-olympics hahstags



Example: #habemuspapam

- PapaFrancesco,francesco,habaemuspapam,habemuspapa,habemuspapam, newpope, papaFrancesco,papafrancesco,pope,Francois1er,HabemusPapam, NouveauPape, habemuspapam,nouveaupape,ConfessionsIntimes, Francesco, HabemusPapa, HabemusPapam,Habemuspapam