Hierarchical Time-Series Clustering for Data Streams

Pedro Rodrigues, João Gama and João Pedro Pedroso

Computer Science Department - Faculty of Science
Artificial Intelligence and Computer Science Laboratory
University of Porto

Overview

- Motivation
- Related Work
- Divisive Analysis Clustering (DIANA)
- Online Divisive-Agglomerative Clustering (ODAC)
  - Behaviour
  - Structure
  - Algorithm Criteria
    - Splitting
    - Split Support
    - Aggregate
- Experimental Work
- Discussion and Future Work
- Summary

Motivation

- Many modern databases consist of continuously stored data from unclassified time-series
  - Power systems, financial market, web logs, network routers, etc...
- Environment is often so dynamic that our models must be always adapting
- Our goal is to design a system that:
  - hierarchically cluster time-series (“whole” clustering);
    - defines clusters of variables
    - one has no need to pre-define the number of clusters
  - is built incrementally, working online;
  - dynamically adapts to new data;
  - basically... works! ;-)

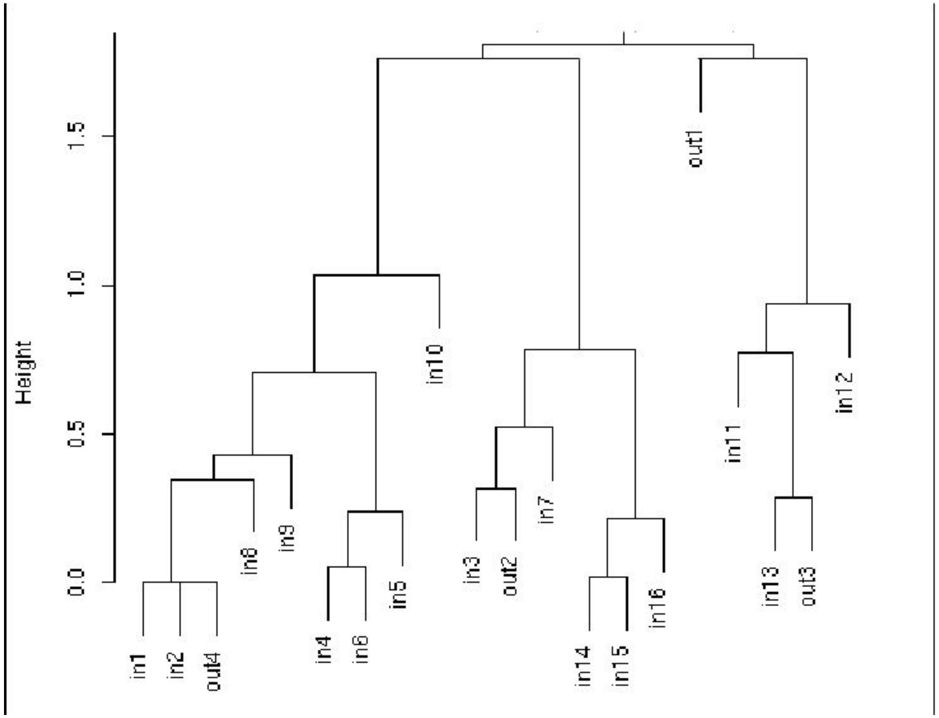
Related Work

- Parametric Clustering
  - Reconstructive models (tend to minimize a cost function)
    - K-means, K-medians, Simulated Annealing, ...
  - Generative models (assume instances are observations from a set of K unknown distributions
    - Gaussian Mixture Model using Expectation-Maximization, C-Means Fuzzy, ...
- Non-parametric Clustering (hierarchical models)
  - usually based on dissimilarities between elements of the same cluster
  - either agglomerative (AGNES) or divisive (DIANA)
- Data Streams
  - VFDT, VFDTc, UFFT, VFML...

Divisive Analysis Clustering (DIANA)

- Starts with one large cluster containing all time-series
- At each step the largest cluster is divided in two
- Stop when all clusters contain only one time-series
- Keep heights of splitting to construct a dendrogram

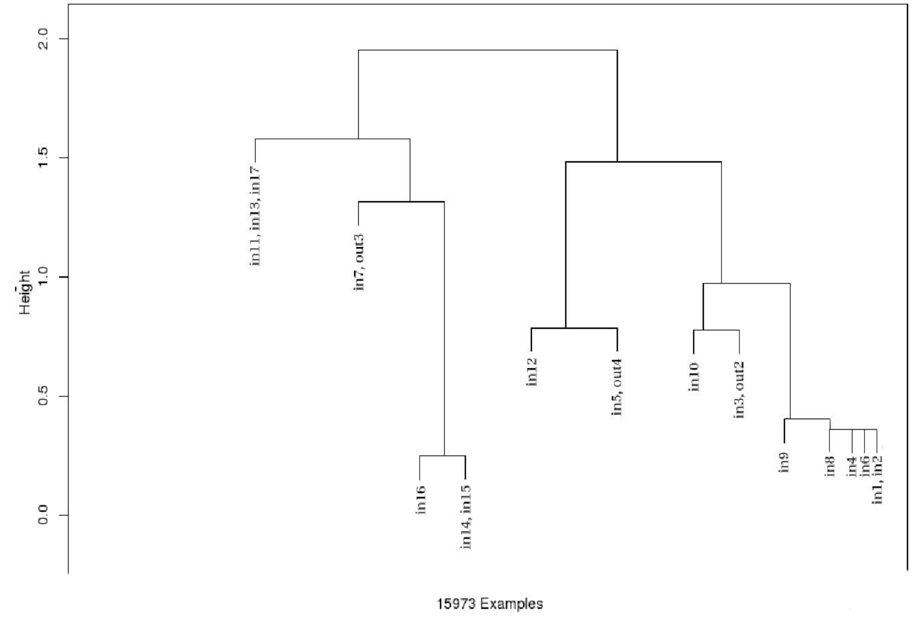
EUNITE dataset
20 variables
15973 examples
1-corr dissimilarity



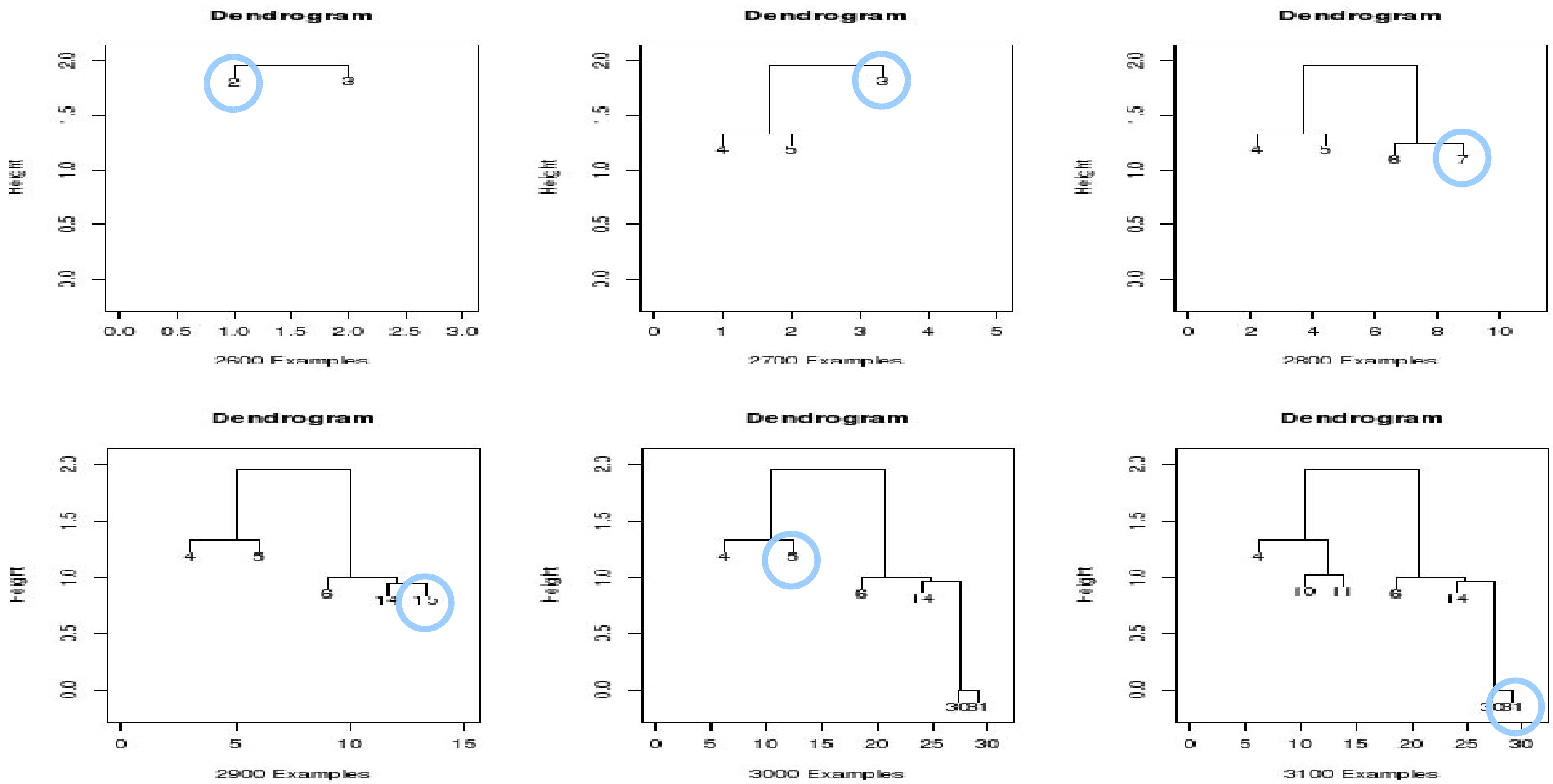
Online Divisive-Agglomerative Clustering (ODAC)

- ODAC main characteristics:
  - Expand Structure
    - divide clusters
  - Contract Structure
    - aggregate clusters
  - Other Issues
    - top-down strategy
    - incremental
    - works online
    - any time cluster definition
    - single scan on data

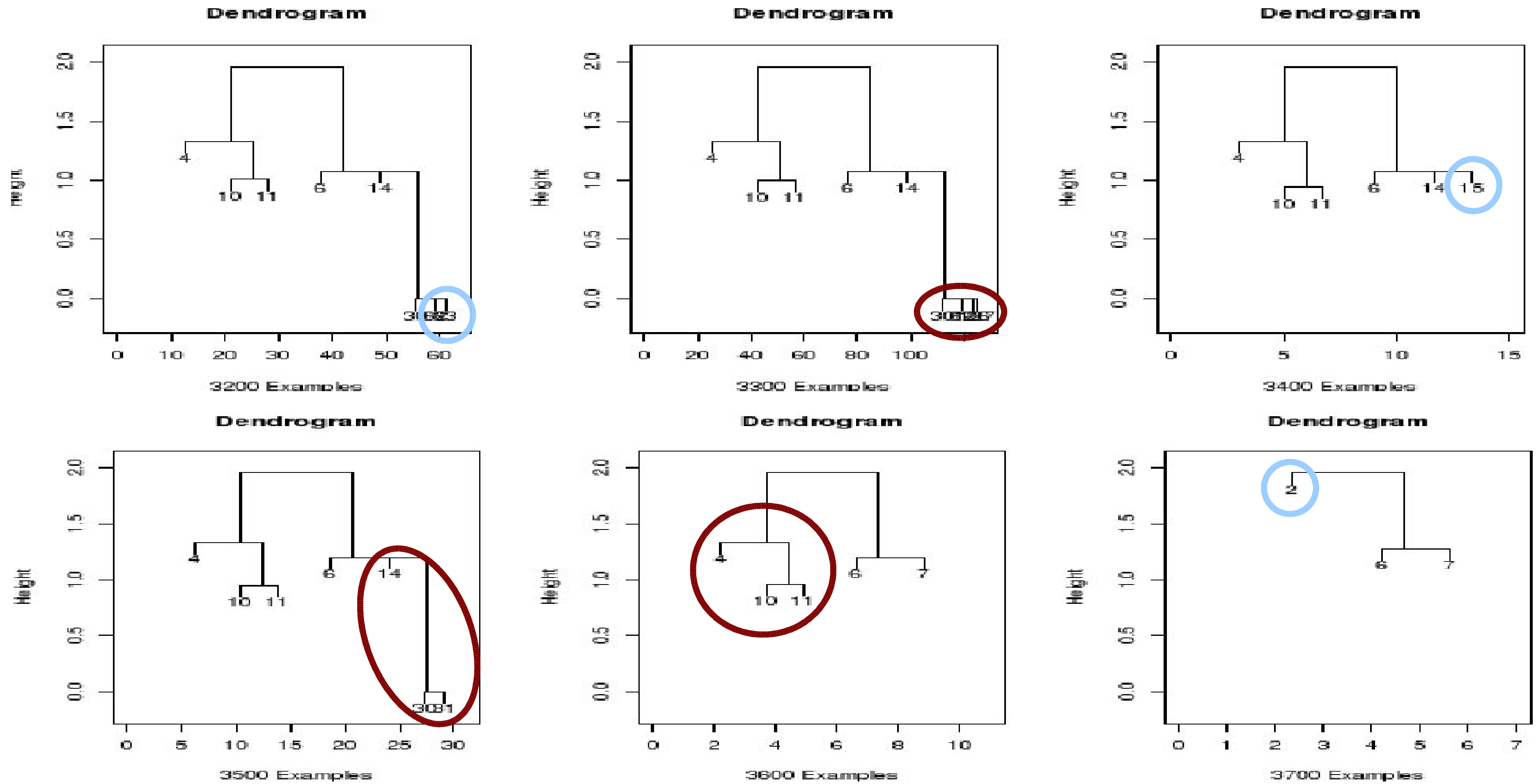
EUNITE dataset
20 variables
15973 examples
1-corr dissimilarity



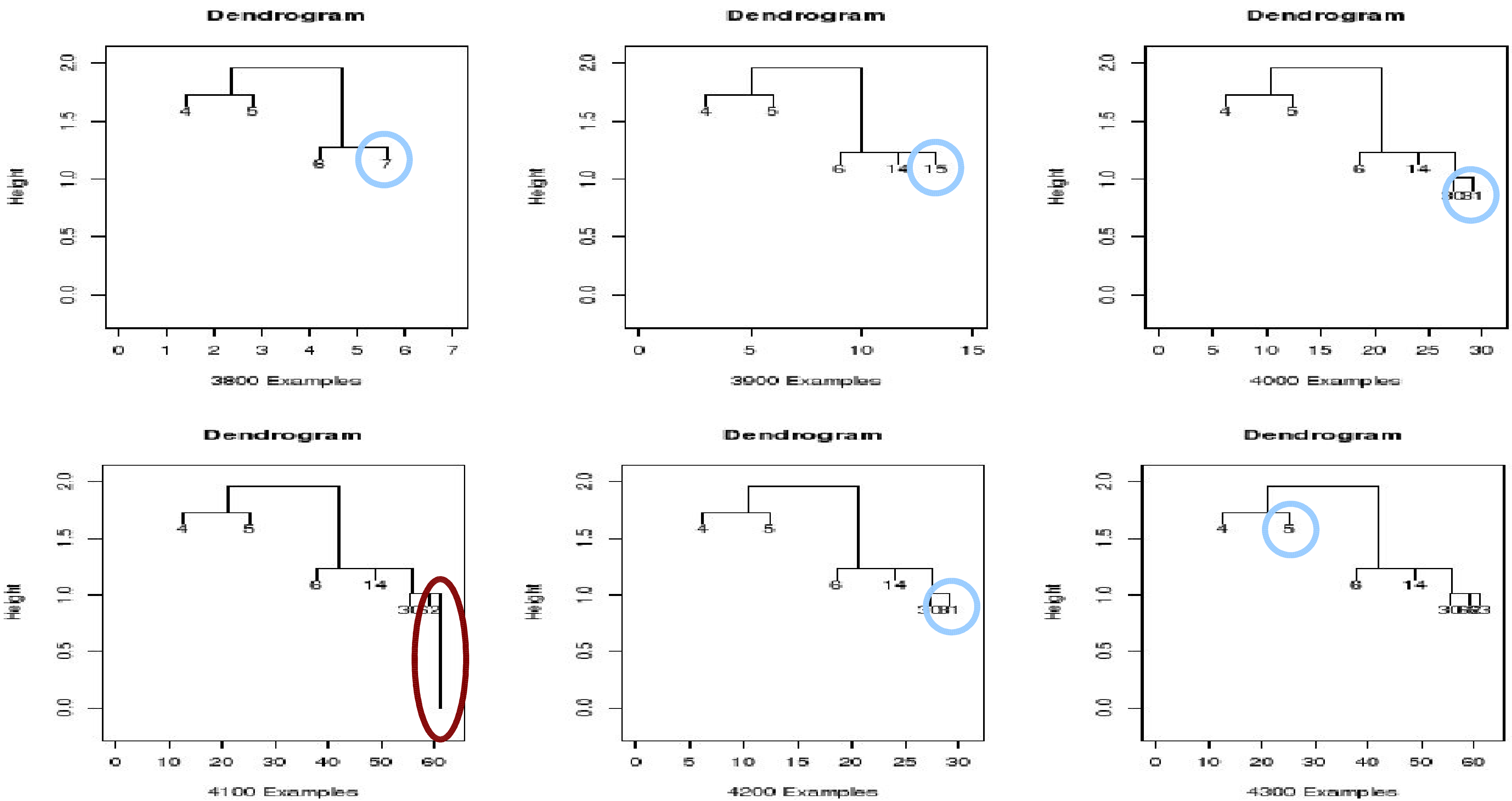
ODAC Behaviour



ODAC Behaviour



ODAC Behaviour



Dissimilarity Measure

- DIANA uses real dissimilarity between time-series

$$d(a,b)=\sum_{i=1}^{n}\frac{|a^i-b^i|}{n}$$

- We could benefit from a ranged measure...

$$corr(a,b)=\frac{\sum_{i=1}^{n}a_i b_i-n\bar{a}\bar{b}}{\sqrt{\sum_{i=1}^{n}a_i^2-n\bar{a}^2}\sqrt{\sum_{i=1}^{n}b_i^2-n\bar{b}^2}}$$

Incremental Correlation

- We can see that the sufficient statistics needed to compute correlation on the fly are...

$$A=\sum_{i=1}^{n} a_i$$

$$B=\sum_{i=1}^{n} b_i$$

$$A2=\sum_{i=1}^{n} a_i^2$$

$$B2=\sum_{i=1}^{n} b_i^2$$

$$AB=\sum_{i=1}^{n} a_i b_i$$

$$N=n$$

$$corr(a,b)=\frac{\sum_{i=1}^{n} a_i b_i - n\bar{a}\bar{b}}{\sqrt{\sum_{i=1}^{n} a_i^2 - n\bar{a}^2}\sqrt{\sum_{i=1}^{n} b_i^2 - n\bar{b}^2}}$$

$$corr_N(a,b)=\frac{AB-\frac{A.B}{N}}{\sqrt{A2-\frac{A^2}{N}}\sqrt{B2-\frac{B^2}{N}}}$$

Dissimilarity Measure - Diameter

- We use as dissimilarity measure between time-series *a* and *b*, at *n* examples:

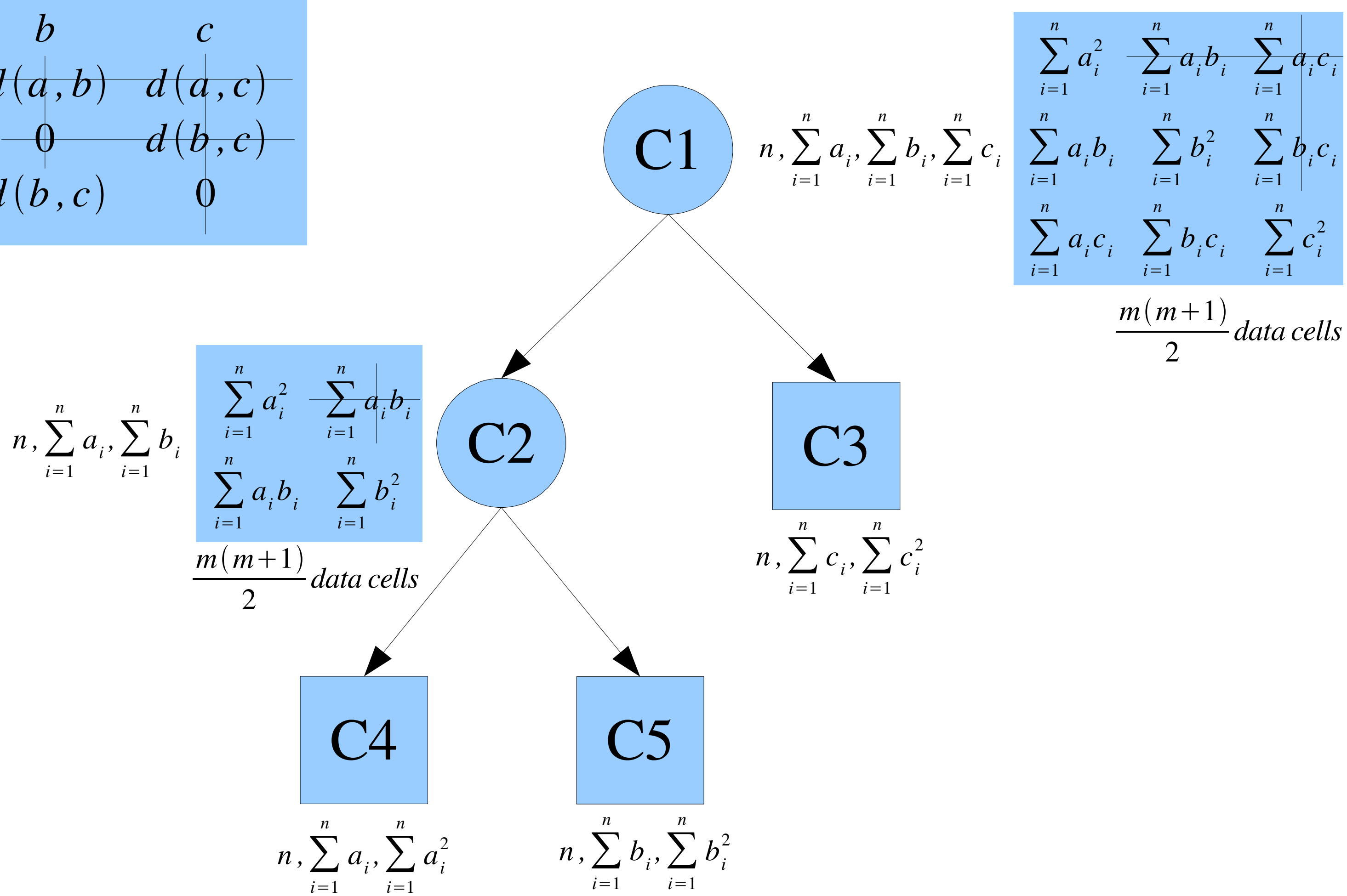
$$d_n : \mathbb{N} \times \mathbb{N} \rightarrow [0,2]_{\mathbb{R}}$$
$$d_n(a, b) = 1 - corr_N(a, b)$$

- As in DIANA, we consider the highest dissimilarity between two time-series belonging to the same cluster as the cluster's *diameter*.

ODAC Structure

	a	b	c
a	0	$d(a,b)$	$d(a,c)$
b	$d(a,b)$	0	$d(b,c)$
c	$d(a,c)$	$d(b,c)$	0

$\frac{m(m-1)}{2}$ *data cells*



Splitting Criteria

- DIANA always splits clusters into single objects
- ODAC splits only when we have confidence on a good decision:
  Hoeffding bound.

- For n independent observations of variable v_k with mean $\bar{v}_k$ and range R, the Hoeffding bound states that with probability $1-\delta$ the true mean of the variable is at least $\bar{v}_k - \epsilon_n$, where

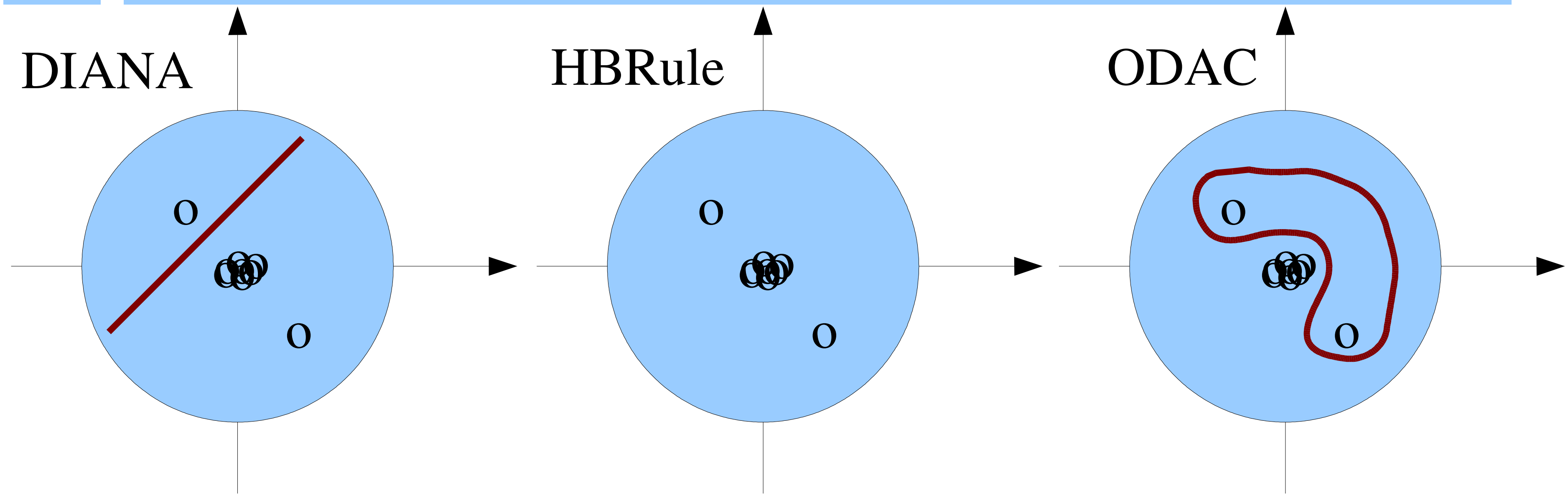
$$\epsilon_n = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

Splitting Criteria

- Let C_k be the largest current cluster on the system. We use an improved version of the splitting rule from DIANA:
- Rank variables $v_i \in C_k$ by average dissimilarity ($\bar{d}_n(i)_k$), in descent order, ex: $$\bar{d}_n(a)_k \geqslant \bar{d}_n(c)_k \geqslant \bar{d}_n(b)_k$$
- Following the Hoeffding bound, we choose to split this cluster if $$\Rightarrow \quad \bar{d}_n(a)_k - \bar{d}_n(c)_k > \epsilon_n$$ ensuring, with confidence $1-\delta$, that this difference is significant.
- But this is not all... what if the two most dissimilar have the same average dissimilarity? The cluster would never be split!

Splitting Criteria Enhanced

Dissimilarity Space View



- If $\bar{d}_n(a)_k - \bar{d}_n(c)_k \leqslant \epsilon_n$ then we test $\bar{d}_k(c) - \bar{d}_k(b) > \epsilon_n$
- If this is true, then we move both variables *a* and *c* to the new cluster and then test for the other variables.
- If not, just follow the ranking until a cut point is found, or no split will occur.

Splitting Criteria Enhanced

- After a split point has been detected, DIANA changes to the new cluster those variables that are closer, in average, to the splinter group then to the remaining group.

$$\bar{d}(b)_k - \bar{d}(b)_s > 0$$

- ODAC has a different perspective:

  Are we confident that the other variables should move to the new cluster along those already moved?

- Move variable b to new cluster C_s if

$$\bar{d}_n(b)_k - \bar{d}_n(b)_s > \epsilon_n$$

Split Support Criteria

- After a split, we only keep the new divided structure if the change really improves a quality measure, the *Divisive Coefficient*.
- Let *dd(i)* be the diameter of the last cluster C_k to which variable v_i belonged, divided by the global diameter. The divisive coefficient of a cluster definition *clust* is

$$DC_{clust} = 2\left(1 - \sum_{i=1}^{m} dd(i)/n\right)$$

- A new cluster definition *clust2* is kept only if

$$DC_{clust2} - DC_{clust} > \epsilon_n$$

ensuring, with confidence $1-\delta$, that the new structure is better (according to the quality measure, of course) than the previous.

Aggregate Criteria

- If at some point, no split occur in any of the existent clusters, we proceed to agglomerative clustering.

- Let C_k be the smallest current cluster of the definition$_{clust}$.
- Assume C_k's parent is a leaf and compute corresponding DC_{clust2}
- Following the Hoeffding bound, we choose this cluster definition if

$$\Rightarrow \quad DC_{clust} - DC_{clust2} \leqslant \epsilon_n$$

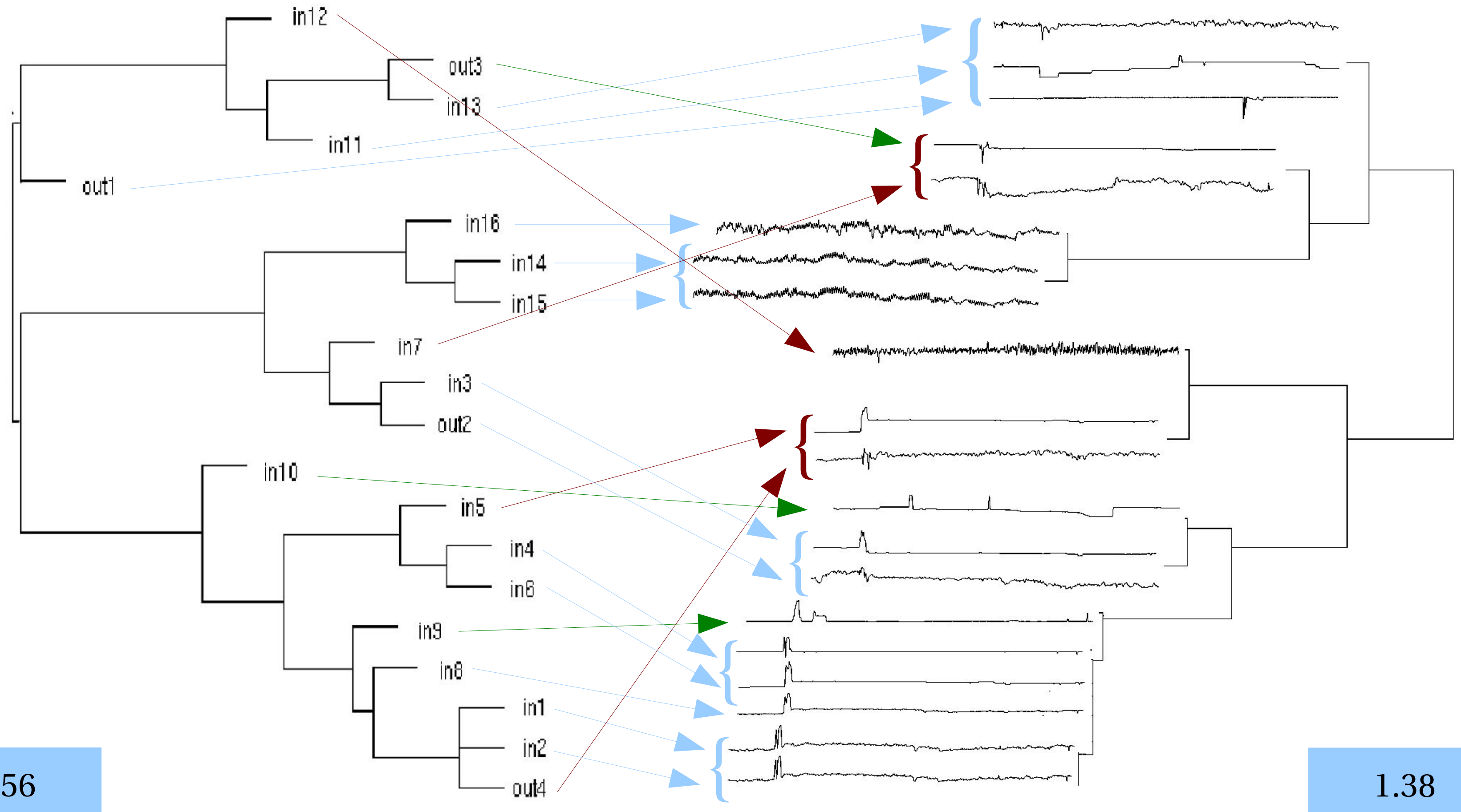
  assuming no confidence that this difference is significant.

- All of C_k's parent's children are pruned!

Experimental Work

DIANA

ODAC



1.56

1.38

Discussion and Future Work

- The system is built incrementally and behaves dynamically;
  Works online, giving an any time cluster definition;

- The system stores all information about variables since it started;
  Concept drift will clarify the notion of learn and forget;

- Some times the system stalls at a given cluster, dividing and aggregating consecutively;

- Benefits from saving computations are still to assure as the divisive coefficient is calculated every time a split support or aggregate decision must be made;

- Introducing new time-series along the iterations is not yet implemented but is on the forge...

Summary

- ODAC - a new algorithm to incrementally cluster online time-series in data streams:
  - hierarchically cluster time-series (“whole” clustering);
  - top-down strategy;
  - is built incrementally, working online;
  - dynamically adapts to new data: dividing and aggregating;
  - any time cluster definition;
  - single scan on data;
  - basically... works ;-)
    ...but should and will (!) work better!

  - What is missing in ODAC?

You tell us, please!

Thank You!

Pedro Rodrigues, João Gama and João Pedro Pedroso

Computer Science Department - Faculty of Science
Artificial Intelligence and Computer Science Laboratory
University of Porto

DIANA Algorithm

1. Select cluster C_k with highest diameter

2. Find the object $s \in C_k$ with highest average dissimilarity to all other objects $i \in C_k$ and start a new cluster C_s (splinter group) with this object.

3. For each object $i \in C_k \setminus C_s$ compute

$$D_i = \{\underset{j}{average}\, d(i,j),\, j \notin C_s\} - \{\underset{j}{average}\, d(i,j),\, j \in C_s\}$$

4. Find the object h with largest difference D_h. If $D_h > 0$ then h is, on average, closer to the splinter group than to the old group, so move h to C_s.

5. Repeat steps 3. and 4. until all D_i are negative

6. If there is a cluster C_k with $\#C_k > 1$ then goto 1.

ODAC Algorithm

1. Get next n_{min} examples

2. Update and propagate sufficient statistics for all variables

3. Compute the Hoeffding bound $(\in)$

4. Choose next cluster C_k in descending order of diameters

5. TestSplit() in cluster C_k

6. If we found a split point, goto 12. with new cluster tree

7. If still exists a cluster C_k not yet tested for splitting goto 4.

8. Choose next cluster C_k in ascending order of diameters

9. TestAggregate() in cluster C_k

10. If we found an aggregation then goto 12. with new cluster tree

11. If still exists a cluster C_k not yet tested for aggregation goto 8.

12. If not end of data, goto 1.