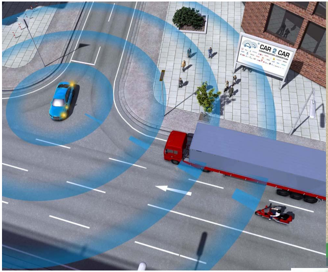
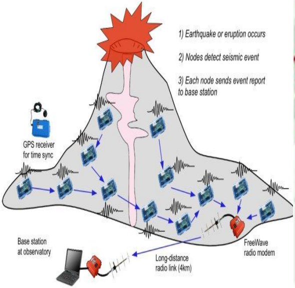
Challenges in Ubiquitous Data Mining

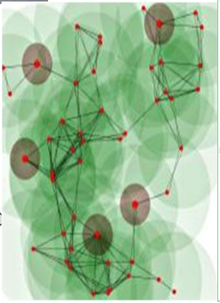
João Gama

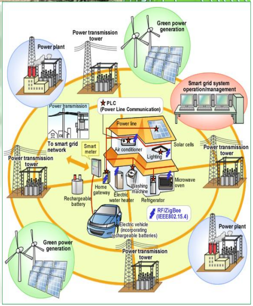
LIAAD-INESC Porto, University of Porto, Portugal
jgama@fep.up.pt

1. Motivation
2. Illustrative Example
   - Very-short-term Forecasting in Photovoltaic Systems
3. Clustering Sensor Networks
   - Motivation
   - Distributed Grid Clustering
   - Clustering Data Sources
4. Final Comments

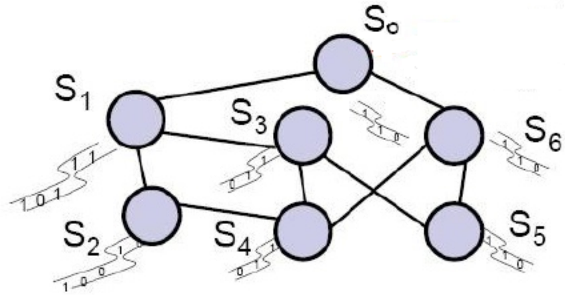

CAR 2 CAR


1) Earthquake or eruption occurs
2) Nodes detect seismic event
3) Each node sends event report to base station
GPS receiver for time sync
Base station at observatory
Long-distance radio link (4km)
FreeWave radio modem




Power plant
Power transmission tower
Green power generation
Smart grid system operation/management
PLC (Power Line Communication)
Power transmission
Power line
To smart grid network
Smart meter
Air conditioner
Solar cells
Lighting
Power transmission tower
Power transmission tower
Home gateway
Electric water heater
Washing machine
Microwave oven
Refrigerator
Rechargeable battery
RF/ZigBee (IEEE802.15.4)
Electric vehicle (incorporating rechargeable batteries)
Green power generation
Power transmission tower
Power plant

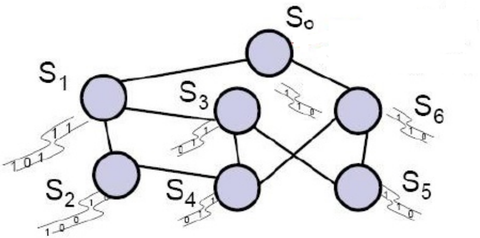
Problem Formulation: Network Data Model



Querying Model

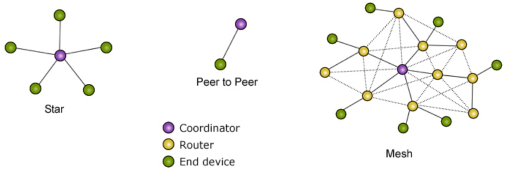
Query $= Q(\bigcup_{i=0}^{n} S_i)$

- One-shot queries:
  What is the state of the network?
- Continuous queries:
  Track and monitor the state of network at any time



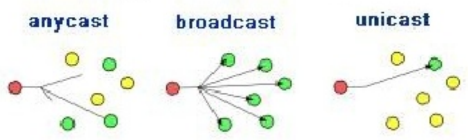
Network topologies

- Star Topology
  arrange peers around a central hub (coordinator).
- Mesh Network
  every peer is connected to nearest peers. The main purpose is fault tolerance.



Routing schemes

- **unicast:** delivers a message to a single specific node;
- **broadcast:** delivers a message to all nodes in the network;
- **anycast:** delivers a message to a group of nodes, typically the ones nearest to the source.



Limitations of existing techniques

- Machine learning so far has mostly centered on one-shot data analysis from homogeneous and stationary data, and on centralized algorithms.
- We are faced with tremendous amount of distributed data.
- In most cases, **this data is transient**, and may not be stored in permanent relations.
- The theory of machine learning relies on the assumption that the data points are independent and identically distributed,
- meaning that the underlying generative process is stationary.

Requirements for Mining Sensor Data Streams

- Vertically distributed data
- Single pass:
  process each observation once;
- Small space:
  constant space;
- Small processing time;
- Reduced communications.

- Local Approaches:
  - ✓ Privacy and Security preserving
  - ✗ Synchronization

The Demand for Learning

Requirements for **adaptive** smart devices:

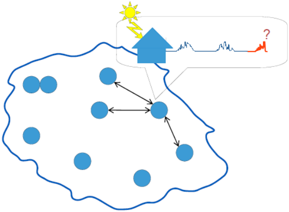
- be able to sense their environment, receive data from other devices, and make sense of the gathered data.
- be able to adapt continuously to **changing environmental conditions** and evolving user habits and needs.
- be capable of **predictive self-diagnosis**.
- be **resource-aware** because of the real-time constraint and of limited computer, battery power and communication resources.

Illustrative Example: Renewable Power Prediction

Analog Method for Collaborative very-short-term Forecasting of Power Generation from Photovoltaic Systems, V.Gomez, G. Hebrail, NGDM 2011

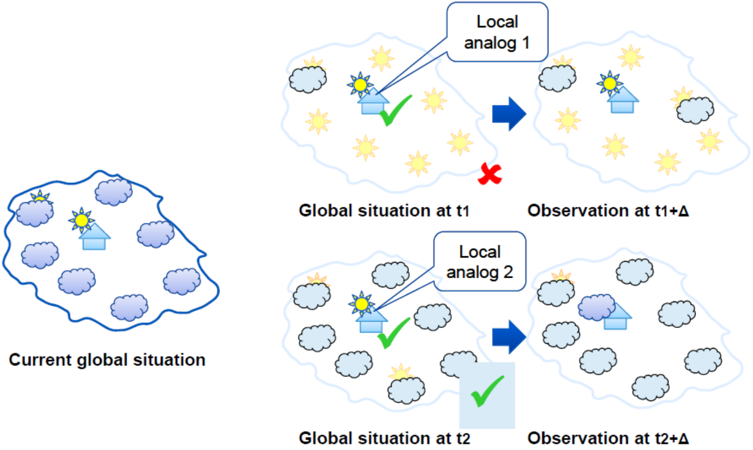
- EC recommendation: in 2020 the penetration of renewable energies should be 20%
- Renewable Power Prediction:
  Predict the power produced by a photovoltaic panel for each quarter in a short-term time horizon.

Collaborative Forecasting: Main Idea

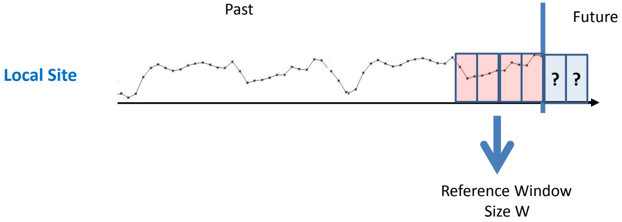


1. **Local Step:** Find past states nearest to current state;
2. **Collaboration:** Broadcast time-stamps of past nearest states;
3. **Local Search:** Inferring the Global Context;
4. **Prediction:** Using the global context.

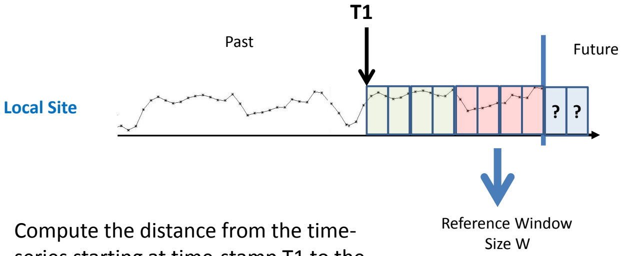
Collaboration



Local Search

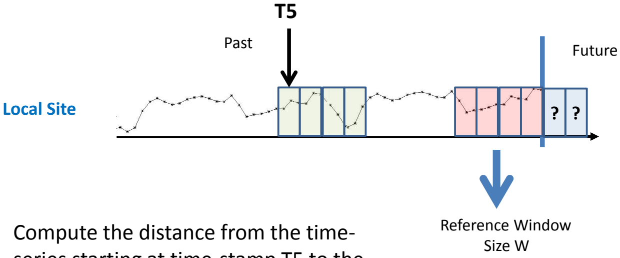


Local Search



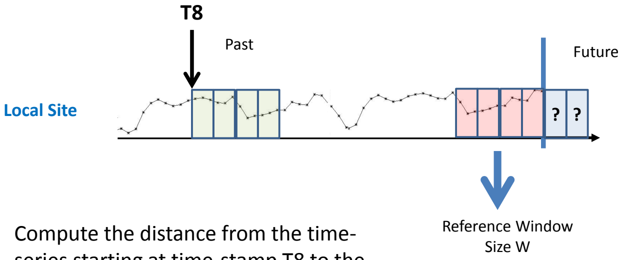
Compute the distance from the time-series starting at time-stamp T1 to the reference window

Local Search



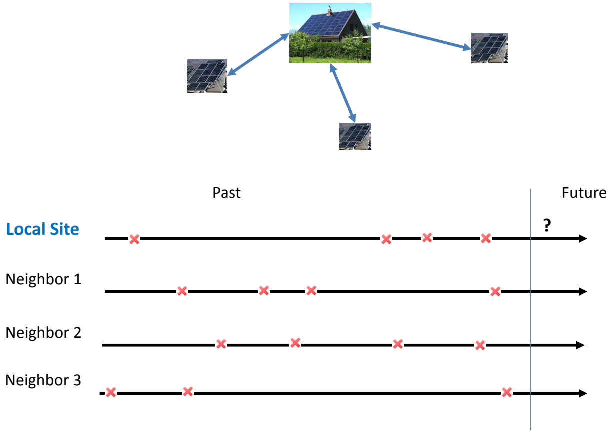
Compute the distance from the time-series starting at time-stamp T5 to the reference window

Local Search

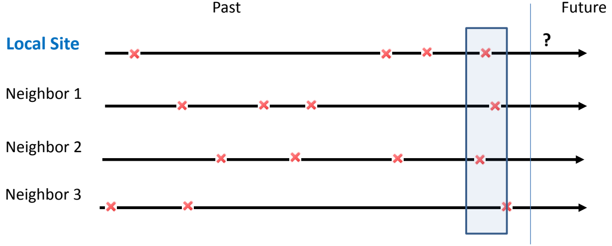


Compute the distance from the time-series starting at time-stamp T8 to the reference window

Collaboration: broadcast time-stamps of similar contexts

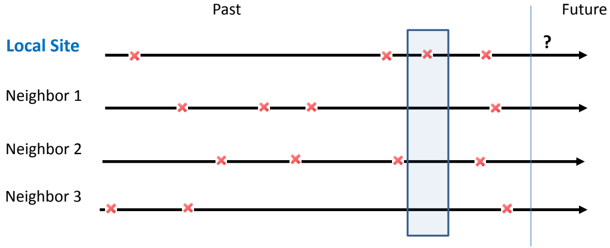


Local search: Inferring the Global Context



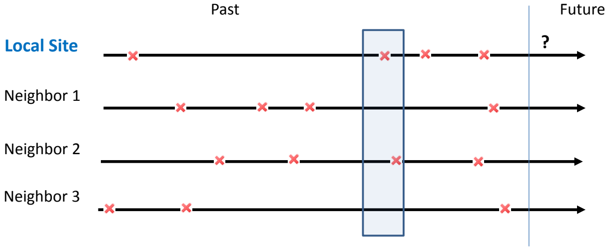
Matches: 3

Local search: Inferring the Global Context



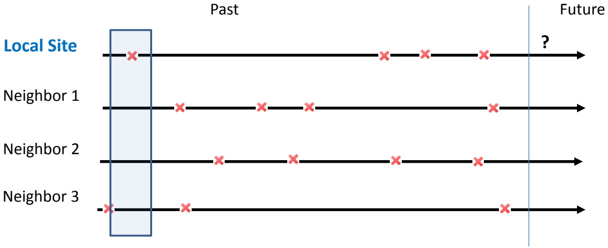
Matches: 0

Local search: Inferring the Global Context



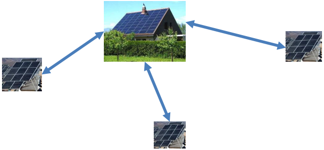
Matches: 1

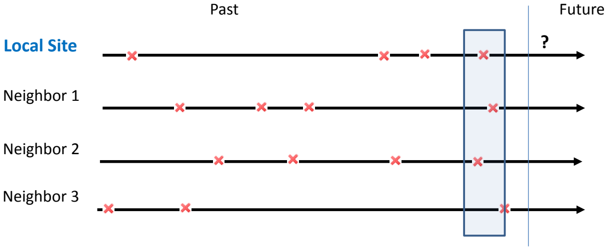
Local search: Inferring the Global Context



Matches: 1

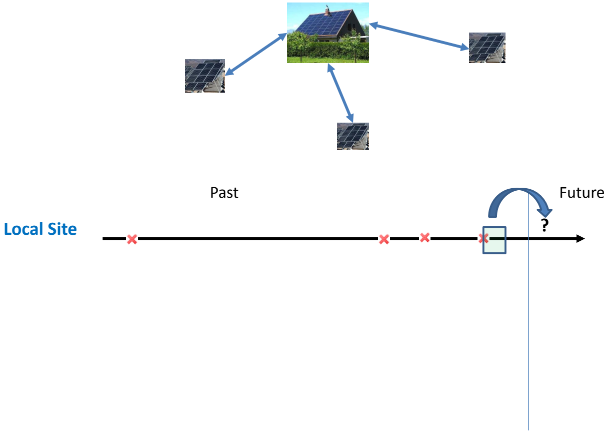
The Global Context





Best Matching: 3

Prediction

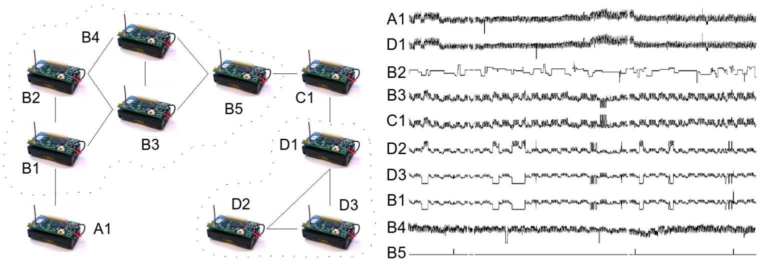


Lessons Learned

- Using local information to infer global context by *collaboration* with neighbors;
- Preserves privacy while collaborating with other systems;

Clustering Distributed Data Streams

Sensors are small, low-cost devices capable of sensing and communicating with other sensors.

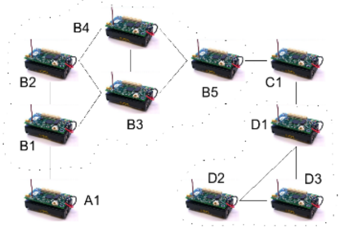


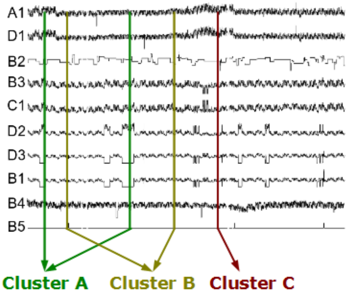
Continuously maintain a cluster structure over the network.

Clustering Distributed Data Streams

Continuously maintain a cluster structure of the data points generated by sensors.

- A Cluster is a set of data points: Information about dense regions of the sensor data space.

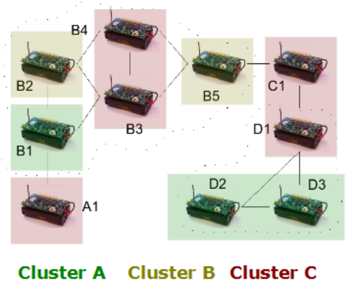


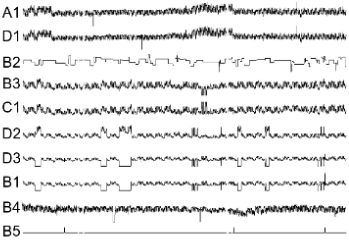


Clustering Distributed Sources of Data Streams

Continuously maintain a cluster structure of the sensors producing data.

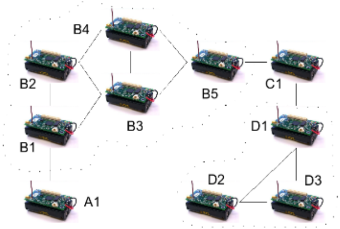
- A Cluster is a set of sensors: Information about groups of sensors that behave similarly over time.

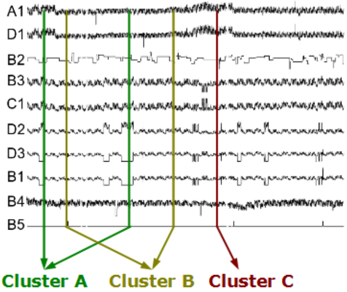




Clustering Distributed Data Streams

- A Cluster is a set of data points.
- Information about dense regions of the sensor data space

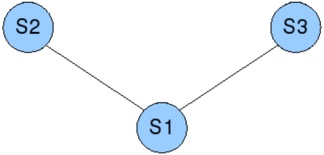




P. Rodrigues, J. Gama: Clustering Distributed Sensor Data Streams. ECML/PKDD 2008

Clustering Distributed Data Streams

Clustering of sensor data gives information about dense regions of the sensor data space.



S1	S2	S3
1	10	102
2	12	110
32	3	44
36	5	36

Roughly speaking, a 2-cluster analysis:

- low S1 ⇔high S2 and S3
- high S1 ⇔ low S2 and S3

Challenges

- High-speed data streams $\rightarrow$ excessive storage and processing;
- Widely spread network $\rightarrow$ heavy communication;
- Centralized clustering $\rightarrow$ high dimensionality;
- Evolving data $\rightarrow$ outdated models;

System Overview

Stpe 1 Each local sensor keeps an online ordinal discretization of its data stream

- Sensor state $\in \{l, m, h\}$;
- Only send state, when it changes.

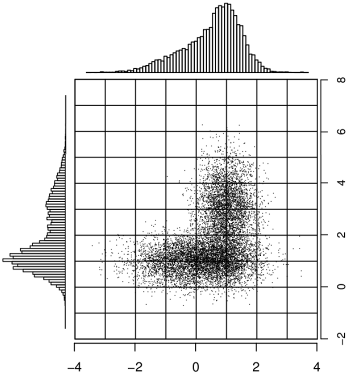
Step 2 The coordinator has the global state of the network

- Network 3 Sensors state $= \{l, l, h\}$;
- keeps a small list of the most frequent states: $\{\langle l, m, h\rangle, \langle l, h, h\rangle \langle m, l, h\rangle, \langle m, l, m\rangle\}$

Step 3 Partitional clustering is applied to the frequent states.

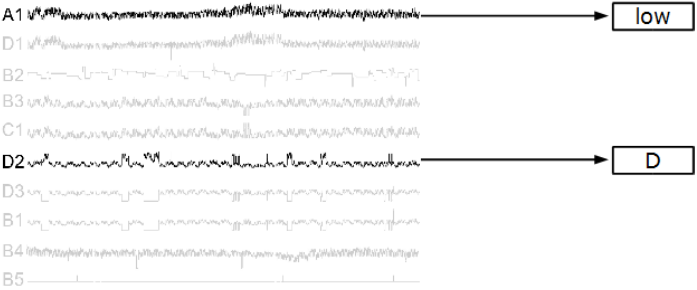
System Overview

Reduce dimensionality and communication



Step 1: Local Step

Each sensor keeps an online discretization of its data.



Reduce dimensionality and communication.

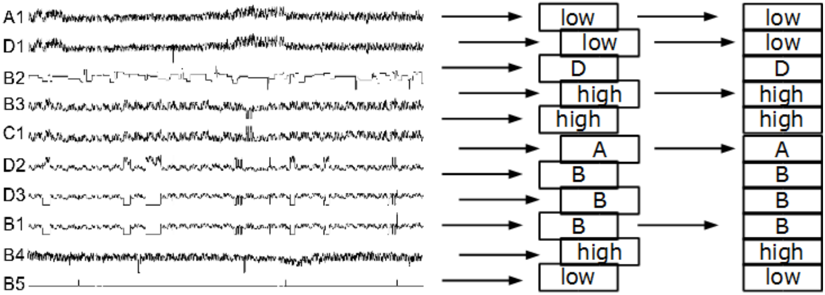
Local Adaptive Grid

- Incremental discretization at each sensor stream X_i using Partition Incremental Discretization ([Gama and Pinto, 2006]).
  - Two layer discretization:
  - The first layer simplifies and summarizes the data, using equal-width discretization;
  - The second layer constructs the final grid by merging the layer-one intervals.
- **Update in constant time and (almost) constant space.**

Step 2: Aggregation Step

The coordinator gathers the global state of the network

Sensors whose state has not changed, do not transmit



Communications

Heavy Load Communication $\Rightarrow$ State sent to coordinator when state changes.

- Each sensor will send its state to the coordinator only if **it has changed** since last communication.
- The **global state** is synchronously updated at each time stamp as a combination of each local site's state; $s(t) = \langle s_1(t); s_2(t); \ldots, s_i(t) \rangle$
- If no information arrives from a local site i, the central site assumes that site i stays in the previous local state: $s_i(t) \leftarrow s : i(t-1)$

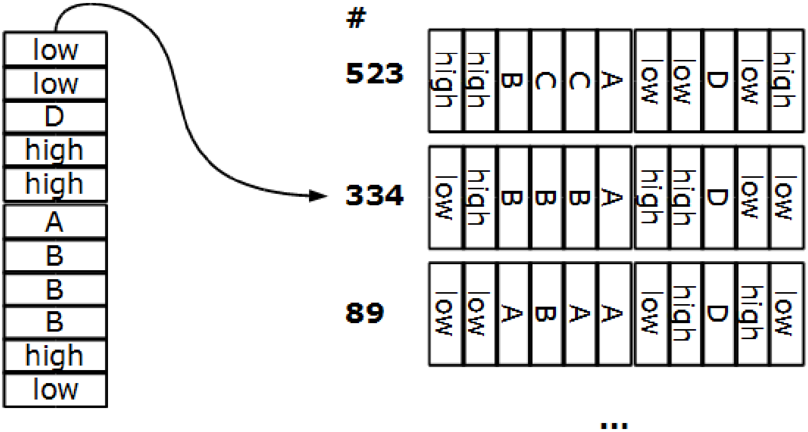
Monitoring States

Metwally, D. , A. Abbadi, *Efficient Computation of Frequent and Top-k Elements in Data Streams*, ICDT 2005

- The number of cell combinations to be monitored by the coordinate site is exponential to the number of sensors: $O(w^d)$.
  Only a small number of them represent frequent states.
- The Space-Saving Algorithm:
  - If current state is being monitored, increment its counter.
  - If it is not being monitored, replace the least frequent monitored state with current state and increment evicted counter.
- **it tends to give more importance to recent examples, enhancing the adaptation of the system to data evolution.**

Frequent States

The coordinator keeps a small list of the most frequent global states



Step 3: Centralized Cluster

Outdated Models $\Rightarrow$ Online Adaptive k-Means Clustering.

- Each frequent state represents a multivariate point, defined by the central points of the corresponding unit cells.
- When the central site has a top-m set of states, with $m > k$, apply a simple partitional algorithm.

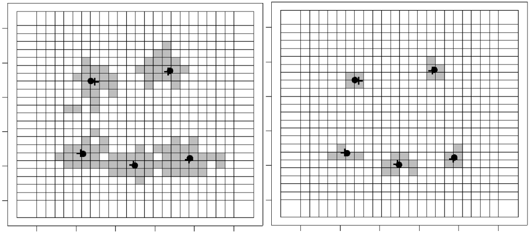
Furthest Point Clustering

Furthest Point clustering:

- the first cluster center c_1 is chosen randomly among data points.
- Subsequent $k-1$ cluster centers are chosen as the points that are more distant from the previous centers c_1, c_2, ..., c_{i-1}, by maximizing the minimum distance to the centers.

Requires k passes over training points.

Illustrative Example



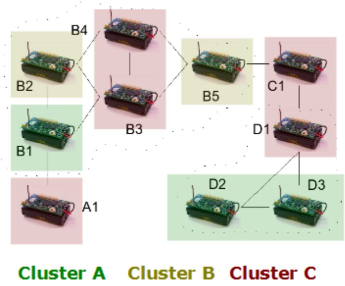
System's granularity can be tuned to the resources available in the network.

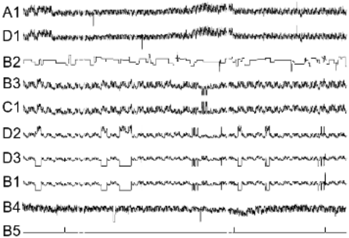
Main Achievements

- Online discretization yields:
  - constant storage and processing load at local sensors;
  - a reduction of the system's sensitivity to uncertainty;
  - a reduction in communication (only when state changes).
- Frequent state monitoring yields:
  - a reduction on the server's memory requirements;
  - definition of representatives of dense regions of the sensor space.
- Online clustering of frequent states yields:
  - a reduction on the number of samples used in clustering;
  - a straightforward adaptation to most recent data.

Clustering Distributed Sources of Data Streams

- A Cluster is a set of sensors;
- Information about groups of sensors that behave similarly over time.





P. Rodrigues, J. Gama: L2GClust: local-to-global clustering of stream sources. SAC 2011

Challenges

P. Rodrigues, J. Gama: L2GClust: local-to-global clustering of stream sources. SAC 2011

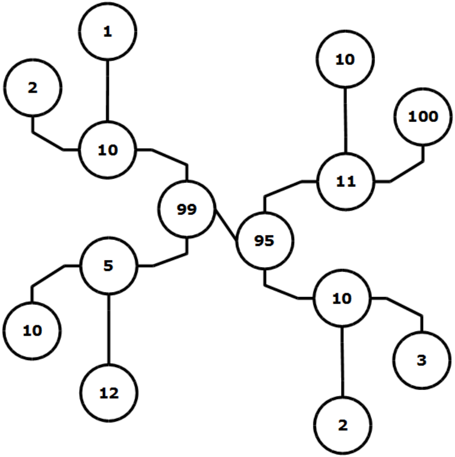
- High-speed data streams → excessive storage and processing;
- Widely spread network → heavy communication;
- Evolving data → outdated models;

A k-means Algorithm for Evolving Data

- Each sensor keeps a sketch of its most recent data.
- Focusing in the most recent data:
  - Sliding windows;
  - Forgetting factors.
- Scarce resources: Memoryless α-fading average

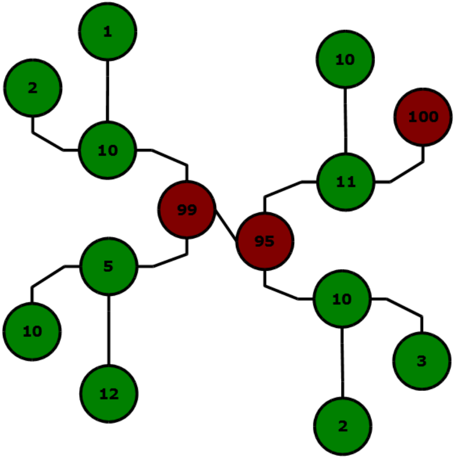
$$M_\alpha(i+1) = \frac{x_i + \alpha \times S_\alpha(i)}{1 + \alpha \times N_\alpha(i)}$$

Example: Local Clustering



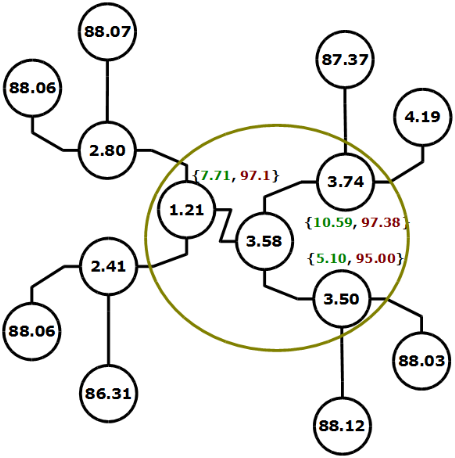
Example: Local Clustering

Centroids $\{6.9, 98.0\}$



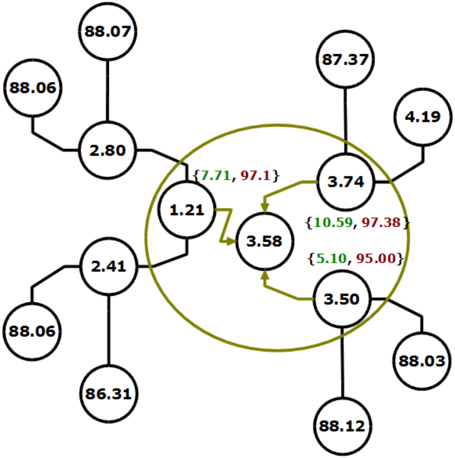
Example: Local Clustering

Centroids {6.9, 98.0}



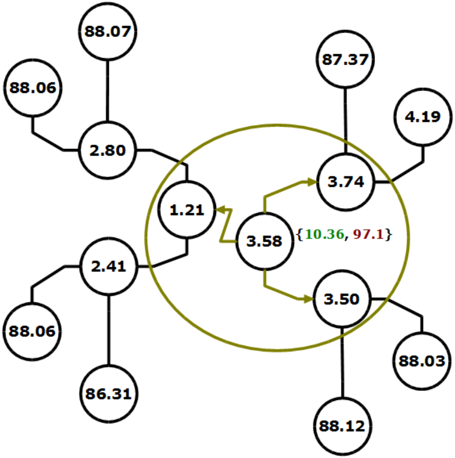
Receiving Neighbors Data

Centroids $\{6.9, 98.0\}$



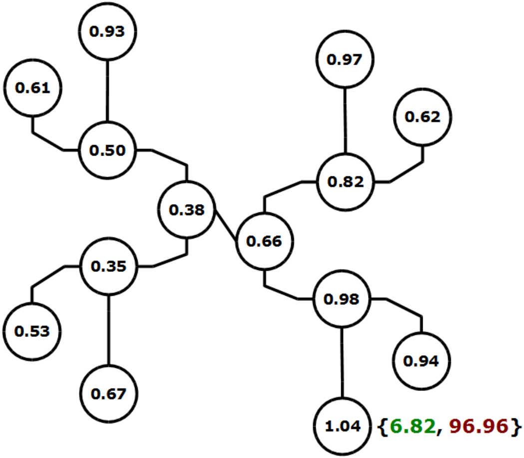
Sending Data to Neighbors

Centroids {6.9, 98.0}



After 512 Iterations...

Centroids $\{6.9, 98.0\}$

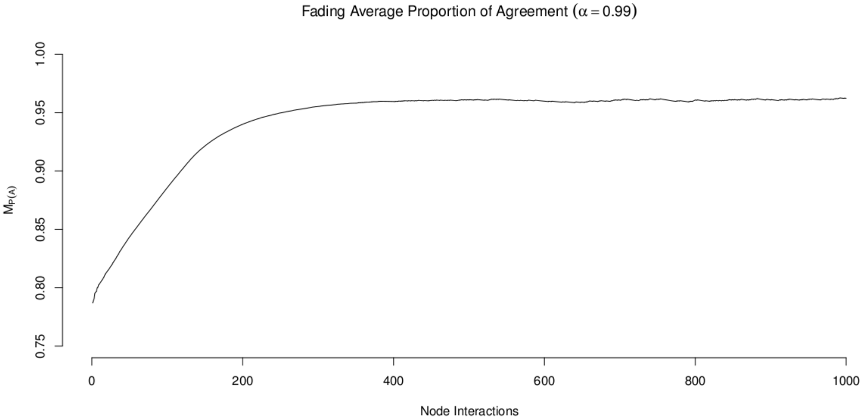


Evaluation

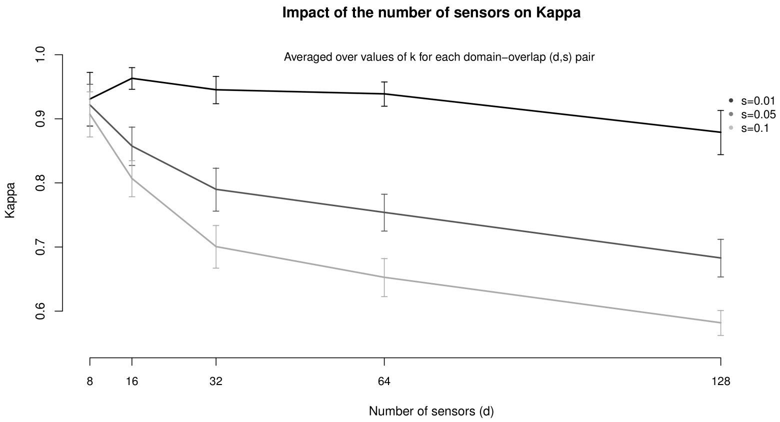
- Cluster validity: Proportion of agreement $P(A)$
- Cluster sanity: Kappa statistic
  $K = (P(A) - P(e))/(1 - P(e))$
  $P(A)$: observed agreement; $P(e)$: agreement by chance
- State-of-the-art Simulator
  Each sensor in the simulation (Visual Sense) generates a Gaussian stream with mean from one of the predefined Gaussian clusters.

Evaluation

Average proportion of agreement converges (with small fluctuations).

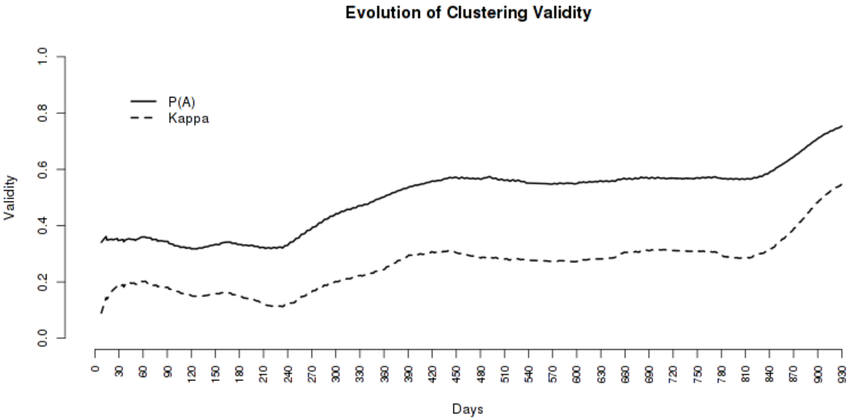


Evaluation


Impact of the number of sensors on Kappa
Averaged over values of k for each domain−overlap (d,s) pair
Kappa
1.0
0.9
0.8
0.7
0.6
s=0.01
s=0.05
s=0.1
8
16
32
64
128
Number of sensors (d)

Evaluation: Electrical Grid Data

Real data from electricity demand sensors



Lessons Learned

- Local sketch yields:
  - memoryless storage of summaries;
  - a straightforward adaptation to most recent data;
  - a reduction of the system's sensitivity to uncertainty;
- Local clustering with direct neighbors yields:
  - no forwarding of information (reduced communication);
  - low dimensionality of the clustering problem;
  - sensitive information better preserved.

A World in Movement

- The new characteristics of data:
  - **Time and space**: The objects of analysis exist in time and space. Often they are able to move.
  - **Dynamic environment**: The objects exist in a dynamic and evolving environment.
  - **Information processing capability**: The objects have limited information processing capabilities
  - **Locality**: The objects know only their local spatio-temporal environment;
  - **Distributed Environment**: Objects will be able to exchange information with other objects.
- Main Goal:
  - **Real-Time Analysis**: decision models have to evolve in correspondence with the evolving environment.

The Challenges of UDM

These characteristics imply:

- Switch from **one-shot learning** to continuously learning **dynamic models** that evolve over time.
- In the perspective induced by ubiquitous environments, *finite training sets, static models, and stationary distributions* will have to be completely thought anew.
- The algorithms will have to use *limited computational resources* (in terms of computations, space and time, communications).

Limited Rationality

Ubiquitous data mining implies new requirements to be considered:

- The algorithms will have to use *limited computational resources* (in terms of computations, space and time).
- The algorithms will have only a *limited random access to data* and may have to communicate with other agents;
- Answers will have to be ready in an *anytime protocol*.
- Data gathering and data (pre-)processing will be *distributed*.
  - *In situ* Data Analysis
  - Think Local – Act Global

Where We Want to Go

The assumption that examples are *independent*, *identically distributed* does not hold.

- Learning in dynamic environments requires *Monitoring the Learning Process*.
- Embedding change detection methods in the learning algorithm is a requirement in the context of continuous flow of data.
- Data is distributed *in nature*:
  - *In situ* Data Analysis
  - Think Local – Act Global

Limited Resources

- The design of learning algorithms must take into account:
  - Memory available is fixed.
  - Computational resources are limited.
  - Communication costs are high.
- Data is distributed *in nature*:
  - *In situ* Data Analysis
  - Think Local – Act Global

Autonomy

Systems and algorithms with high level of autonomy:

- These systems address the problems of data processing, modeling, prediction, clustering, and control in changing and evolving environments.
- They self-evolve their structure and knowledge about the environment.
- They self-monitor the evolution of the learning process.

Thank you!