Scaling Geometric Monitoring Over Distributed Streams

Alexandros D. Keros

June 23, 2016

Supervised by: Prof. V.Samoladas

Table of contents

Data Stream Systems

- **Data streams**: Continuous, high volume, size unbound, violative, probably distributed
- *Pull paradigm*
- Centralizing and/or polling $\rightarrow$ prohibitive in terms of communication overhead
- Examples: telecommunication, sensor networks

The Geometric Monitoring Method

- Threshold monitoring
- Nodes communicate when needed
  - Local constraints
  - Violation resolution (*false alarms*)
- Arbitrary function monitoring
- Tight accuracy bounds
- A promising framework for *distributed data stream monitoring*

Motivation

Problems:

- increasing node population
- data volume
- data dimensionality
- arbitrary functions
- **communication - accuracy tradeoff**

Need for:

- scalability warranties
- tight accuracy bounds
- incremental/real-time operation
- **Minimize communication while retaining accuracy bounds**

Motivation

Problems:

- increasing node population
- data volume
- data dimensionality
- arbitrary functions
- **communication - accuracy tradeoff**

Need for:

- scalability warranties
- tight accuracy bounds
- incremental/real-time operation
- **Minimize communication while retaining accuracy bounds**

Motivation

Problems:

- increasing node population
- data volume
- data dimensionality
- arbitrary functions
- **communication - accuracy tradeoff**

Need for:

- scalability warranties
- tight accuracy bounds
- incremental/real-time operation
- **Minimize communication while retaining accuracy bounds**

Motivation

Problems:

- increasing node population
- data volume
- data dimensionality
- arbitrary functions
- **communication - accuracy tradeoff**

Need for:

- scalability warranties
- tight accuracy bounds
- incremental/real-time operation
- **Minimize communication while retaining accuracy bounds**

Contributions

Expand the *geometric monitoring method*:

- heuristic method for violation resolution
- distance-based hierarchical node clustering
- throughout method evaluation on synthetic and real-world datasets

Contributions

Expand the *geometric monitoring method*:

- heuristic method for violation resolution
- distance-based hierarchical node clustering
- throughout method evaluation on synthetic and real-world datasets

Contributions

Expand the *geometric monitoring method*:

- heuristic method for violation resolution
- distance-based hierarchical node clustering
- throughout method evaluation on synthetic and real-world datasets

Geometric Threshold Monitoring

- **Threshold monitoring**: arbitrary function $f(\cdot)$, threshold T

$$f(\cdot) < T \text{ or } f(\cdot) > T$$

- **Idea**: decompose into local constraints at the nodes

System Architecture

Decentralized Scenario

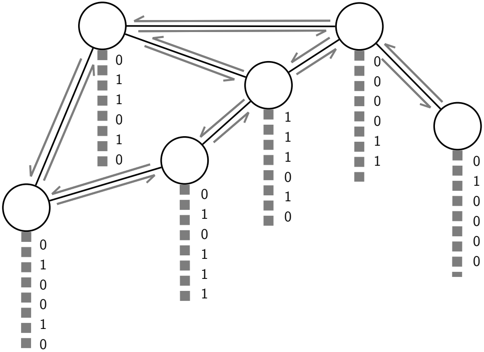


Figure: **Mesh-like network topology** example of the decentralized scenario. Dashed lines represent data streams and half arrows represent message exchanges.

System Architecture

Centralized Scenario

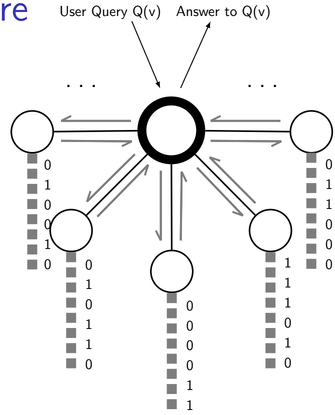


Figure: **Star-like network topology** example of the centralized scenario.The bold node represents the coordinator node. Dashed lines represent data streams and half arrows represent message exchanges.

Computational Model

Statistics vectors

- the *monitoring function* $f : \mathbb{R}^d \rightarrow \mathbb{R}$
- the *threshold* $T \in \mathbb{R}$
- the *monitoring node set* : $P = \{p_1, \ldots, p_n\}$
  with *weights* $w_1, \ldots, w_n$
- the *data streams* : $S = \{s_1, \ldots, s_n\}$
- the d-dimensional *local statistics vectors* : $\vec{v_1}(t), \ldots, \vec{v_n}(t)$
  represent each node's data stream at time t

Global statistics vector

$$\vec{v}(t) = \frac{\sum_{i=1}^{n} w_i \vec{v_i}(t)}{\sum_{i=1}^{n} w_i} \quad (1)$$

Computational Model

Estimate vector

Infrequent communication between nodes/nodes-coordinator:

Estimate vector

$$\vec{e}(t) = \frac{\sum_{i=1}^{n} w_i \vec{v_i}'}{\sum_{i=1}^{n} w_i} \quad (2)$$

- the last communicated *local statistics vector* of node p_i : $\vec{v_i}'$
- *Local statistics* divergence:$\Delta\vec{v_i}(t) = \vec{v_i}(t) - \vec{v_i}', i = 1, \ldots, n$

Decentralized drift vector

$$\vec{u_i}(t) = \vec{e}(t) + \Delta\vec{v_i}(t) \quad (3)$$

Centralized drift vector

$$\vec{u_i}(t) = \vec{e}(t) + \Delta\vec{v_i}(t) + \frac{\vec{\delta_i}}{w_i} \quad (4)$$

Computational Model

Balancing Process

Centralized scenario

Purpose: resolve possible false alarms

Balancing vector

$$\vec{b} = \frac{\sum_{p_i \in P'} w_i \vec{u_i}(t)}{\sum_{p_i \in P'} w_i} \tag{5}$$

- the *balancing set* P': a subset of nodes
- the *slack vector* at the nodes $\vec{\delta_i} = \vec{\delta_i}' + \Delta\vec{\delta_i}$, $\sum_{p_i \in P'} \Delta\vec{\delta_i} = \vec{0}$:

$$\Delta\vec{\delta_i} = w_i \vec{b} - w_i \vec{u_i}(t) \ \forall \ p_i \in P' \tag{6}$$

, readjusts the *drift vectors* (4).

Geometric Interpretation

Convexity Property

Convexity Property

$$\vec{v}(t) = \frac{\sum_{i=1}^{n} w_i \vec{u_i}(t)}{\sum_{i=1}^{n} w_i} \tag{7}$$

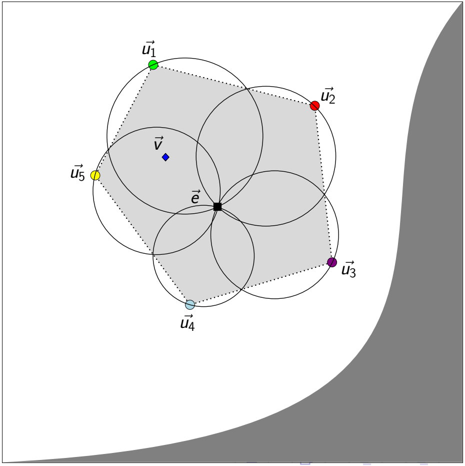
Theorem (Sharfman et al. [3])

Let $\vec{x}, \vec{y_1}, \ldots, \vec{y_n} \in \mathbb{R}^d$ be a set of vectors in $\mathbb{R}^d$. Let $Conv(\vec{x}, \vec{y_1}, \ldots, \vec{y_n})$ be the convex hull of $\vec{x}, \vec{y_1}, \ldots, \vec{y_n}$. Let $B(\vec{x}, \vec{y_i})$ be a ball centered at $\frac{\vec{x}+\vec{y_i}}{2}$ and with radius of $\|\frac{\vec{x}+\vec{y_i}}{2}\|_2$ i.e., $B(\vec{x}, \vec{y_i}) = \{\vec{z} \mid \|\vec{z} - \frac{\vec{x}+\vec{y_i}}{2}\|_2 \leq \|\frac{\vec{x}+\vec{y_i}}{2}\|_2\}$, then $Conv(vecx, \vec{y_1}, \ldots, \vec{y_n}) \subset B(\vec{x}, \vec{y_i})$.

Geometric Interpretation

Convexity Property & Local Constraints

Figure: Example of a convex hull (light gray) defined by the drift vectors $\vec{u_i}, i = 1, 2, 3, 4, 5$. The hull is bounded by the spheres created from the estimate vector $\vec{e}$ and the drift vectors $\vec{u_i}, i = 1, 2, 3, 4, 5$. The global statistics vector $\vec{v}$ is guaranteed to be contained in the convex hull of the drift vectors.



Protocol

Decentralized Algorithm

Algorithm 1: Decentralized algorithm

begin
  foreach *node* p_i **do** /* Node initialization */
    Broadcast $\vec{v}_i(0)$;
    $\vec{v}_i{}' = \vec{v}_i(0)$;
    Wait messages from all other nodes;
    if *messages from all vectors received* **then**
      $\vec{e}(t) = \frac{\sum_{i=1}^{n} w_i \vec{v}_i{}'}{\sum_{i=1}^{n} w_i}$;
    end
  end
  foreach *node* p_i **do** /* Main monitoring task */
    foreach *new* s_i *stream update* $\vec{v}_i(t)$ **do**
      Recalculate $\vec{u}_i(t) = \vec{e}(t) + \Delta\vec{v}_i(t)$;
      if $B(\vec{e}, \vec{u}_i(t))$ *is not monochromatic* **then**
        Broadcast message $< i, \vec{v}_i(t) >$;
        Set $\vec{v}_i{}' = \vec{v}_i(t)$;
      end
      if *new message* $< j, \vec{v}_j(t) >$ *received* **then**
        Set $\vec{v}_j{}' = \vec{v}_j(t)$;
        Recalculate $\vec{e}(t) = \frac{\sum_{i=1}^{n} w_i \vec{v}_i{}'}{\sum_{i=1}^{n} w_i}$;
        if $B(\vec{e}, \vec{u}_i(t))$ *is not monochromatic* **then**
          Broadcast message $< i, \vec{v}_i(t) >$;
          Set $\vec{v}_i{}' = \vec{v}_i(t)$;
        end
      end
    end
  end
end

Protocol

Centralized Algorithm

Algorithm 2: Centralized algorithm's coordinator node operation

begin
  Wait for $< INIT, \cdot >$ messages from all monitoring nodes; `/* Initialization */`
  $\vec{e}(0) = \frac{\sum_{i=1}^{n} w_i \vec{v_i}(0)}{\sum_{i=1}^{n} w_i}$;
  if *new* $< REP, \vec{v_i}(t), \vec{u_i}(t) >$ *message received* **then** `/* Monitoring operation */`
    $P' = P' \cup \{< i, \vec{v_i}(t), \vec{u_i}(t) >\}$;
    `Balance`(P');
  end
end

Function `Balance`(P') `/* Balancing Process */`
  $\vec{b} = \frac{\sum_{p_i \in P'} w_i \vec{u_i}(t)}{\sum_{p_i \in P'} w_i}$;
  if $B(\vec{e}, \vec{b})$ *is not monochromatic* **then**
    if $P - P' \neq \emptyset$ **then**
      Send $< REQ >$ message to random node in $P - P'$ set;
    else
      $\vec{e}(t) = \frac{\sum_{i=1}^{n} w_i \vec{v_i}(t)}{\sum_{i=1}^{n} w_i}$;
      Send $< NEW\text{-}EST, \vec{e}(t) >$ message to all nodes;
      return;

Algorithm 3: Centralized algorithm's monitoring node operation

begin
  foreach *node* p_i **do** `/* Node initialization */`
    Send $< INIT, \vec{v_i}(0) >$ message to coordinator;
    $\vec{v_i}' = \vec{v_i}(0)$;
    $\vec{\delta_i} = \vec{0}$;
    Wait message from coordinator;
    if $< NEW\text{-}EST, \vec{e} >$ *message received* **then**
      Set $\vec{e}(t) = \vec{e}$;
    end
  end
  foreach *node* p_i **do** `/* Main monitoring task */`
    foreach *new* s_i *stream update* $\vec{v_i}(t)$ **do**
      Recalculate $\vec{u_i}(t) = \vec{e}(t) + \Delta\vec{v_i}(t) + \frac{\vec{\delta_i}}{w_i}$;
      if $B(\vec{e}, \vec{u_i}(t))$ *is not monochromatic* **then**
        Send $< REP, \vec{v_i}(t), \vec{u_i}(t) >$ message to coordinator;
        Wait for $< NEW\text{-}EST, \cdot >$ or $< ADJ\text{-}SLK, \cdot >$ message from coordinator;
      end
      if *new message* $< REQ >$ *received* **then**
        Send $< REP, \vec{v_i}(t), \vec{u_i}(t) >$ message to coordinator;
        Wait for $< NEW\text{-}EST, \cdot >$ or $< ADJ\text{-}SLK, \cdot >$ message from

Multi-objective Optimization

Non-linear Constraint Optimization

Primal Descent

Feasible Directions

SQP

The Savitzky-Golay Filter

Maximum Weight Matching

The Primal-Dual Method

Related Work

Problem Formulation

The Geometric Monitoring Framework

The Distance-based Hierarchical Clustering

The Idea

The Distance-based Hierarchical Clustering

The Weight Function

The Distance-based Hierarchical Clustering

The Algorithm

The Heuristic Balancing

The Idea

The Heuristic Balancing

The Optimizing Function

The Heuristic Balancing

The Function Formulation

The Heuristic Balancing

The Algorithm

An Nested Optimization Problem

Velocity and Acceleration Estimation via SG Filtering

Implementation Challenges

Synthetic Data

Real-world Data

Notation

RAND, DIST, DISTR Comparison

GM, HM Comparison

GM, HDM Comparison

Synthetic Data Monitoring

GM, HDM Comparison

Air Pollution Monitoring

Summary & Concluding Remarks

Future Work

The end

Questions?