Lightweight Monitoring of Distributed Data Streams

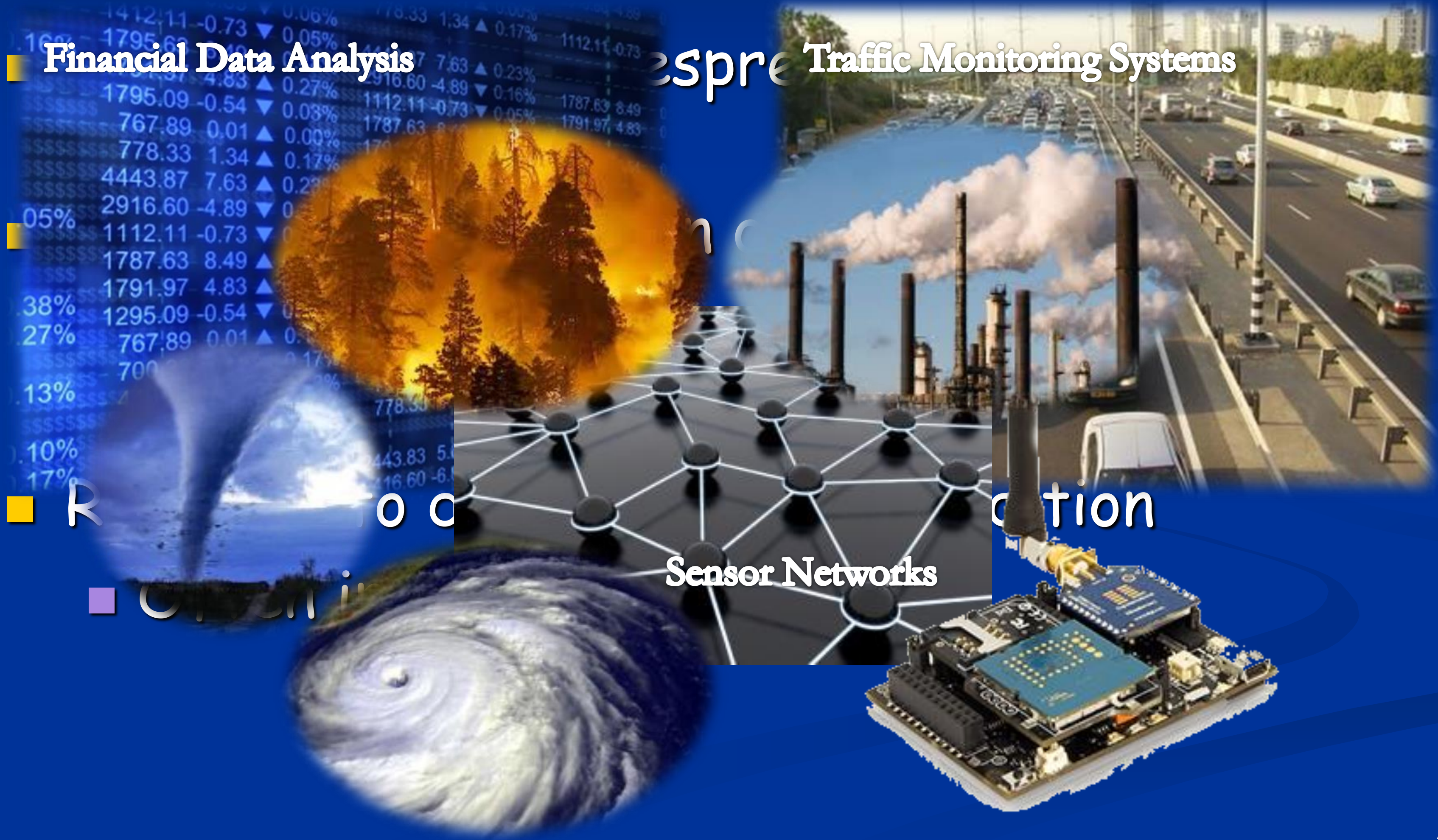
ARNON LAISERSON

DANNY KEREN

ASSAF SCHUSTER

The research leading to these results has received funding from the European Union's Seventh Framework Programme FP7-ICT-2013-11 under grant agreement No 619491 and No 619435.

Distributed Stream Networks



Air Quality Monitoring

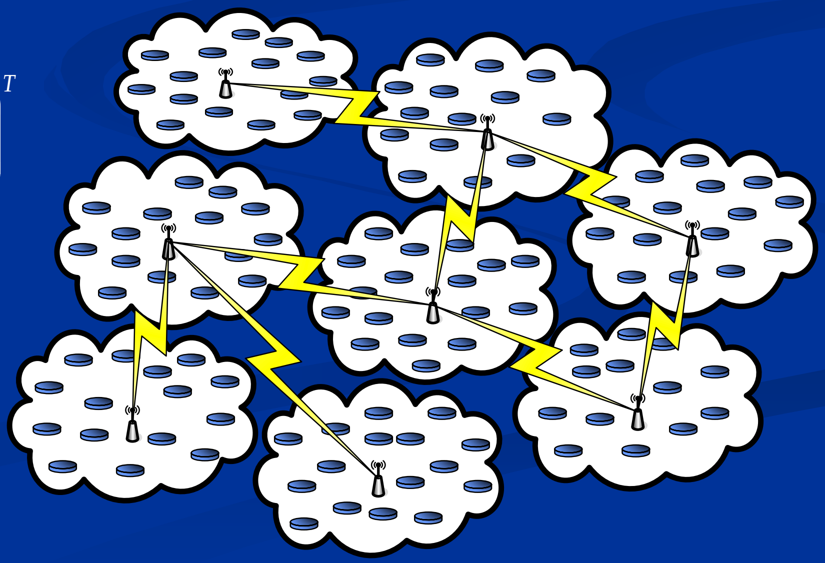
- Sensors monitoring the concentration of air pollutants.

- Each sensor holds a data vector comprising measured concentration of various pollutants (NO, NO_2, CO, CO_2, SO_2, O_3, etc.).
- A function on the *average* readings determines the Air Quality Index (AQI)
- Issue an alert in case the AQI exceeds a given threshold.

Sensor Networks

- Sensors monitoring the temperature in a server room (machine room, conference room, etc.)
  - Ensure uniform temp.: monitor variance of readings
  - Alert in case variance exceeds a threshold
- Temperature readings by *n* sensors $x_1, \ldots, x_n$
- Each sensor holds a data vector $v_i = (x_i^2, x_i)^T$
- The *average* data vector is $v = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2 \quad \frac{1}{n}\sum_{i=1}^{n} x_i\right)^T$
- *Var*(all sensors) =

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2$$



Search Engine

- Distributed datacenter/warehouse
  - "Our logs are larger than any other data by orders of magnitude. They are our source of truth." Sridhar Ramaswamy. **SIGMOD'08 keynote on "Extreme Data Mining"**
- Monitoring the logs: "for which pairs of keywords the correlation index becomes high?"
  - Can change in seconds
- Thousands simultaneous tasks
  - "Network bandwidth is a relatively scarce resource in our computing environment". Dean and Ghemawat. **MapReduce paper, OSDI'04**

Cloud Health Monitoring

amazon webservices SERVICE HEALTH DASHBOARD

Amazon Web Services » Service Health Dashboard

Amazon S3 Availability Event: July 20, 2008

Amazon S3 Availability Event: July 20, 2008

"At 8:40am PDT, error rates in all Amazon S3 datacenters began to quickly climb and our alarms went off. By 8:50am PDT, error rates were significantly elevated and very few requests were completing successfully. By 8:55am PDT, we had multiple engineers engaged and investigating the issue. Our alarms pointed at problems processing customer requests in multiple places within the system and across multiple data centers. While we began investigating several possible causes, we tried to restore system health... At 9:41am PDT, we determined that servers within Amazon S3 were having problems... By 11:05am PDT, all server-to-server communication was stopped, request processing components shut down, and the system's state cleared.... "

Cloud Health Monitoring – Take 2

amazon webservices SERVICE HEALTH DASHBOARD

Amazon Web Services » Service Health Dashboard

Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region

April 29, 2011

Now that we have fully restored functionality to all affected services, we would like to share more details with our customers about the events that occurred with the Amazon Elastic Compute Cloud ("EC2") last week, our efforts to restore the services, and what we are doing to prevent this sort of issue from happening again. We are very aware that many of our customers were significantly impacted by this event, and as with any significant service issue, our intention is to share the details of what happened and how we will improve the service for our customers.

The issues affecting EC2 customers last week primarily involved a subset of the Amazon Elastic Block Store ("EBS") volumes in a single Availability Zone within the US East Region that became unable to service read and write operations. In this document, we will refer to these as "stuck" volumes. This caused instances trying to use these affected volumes to also get "stuck" when they

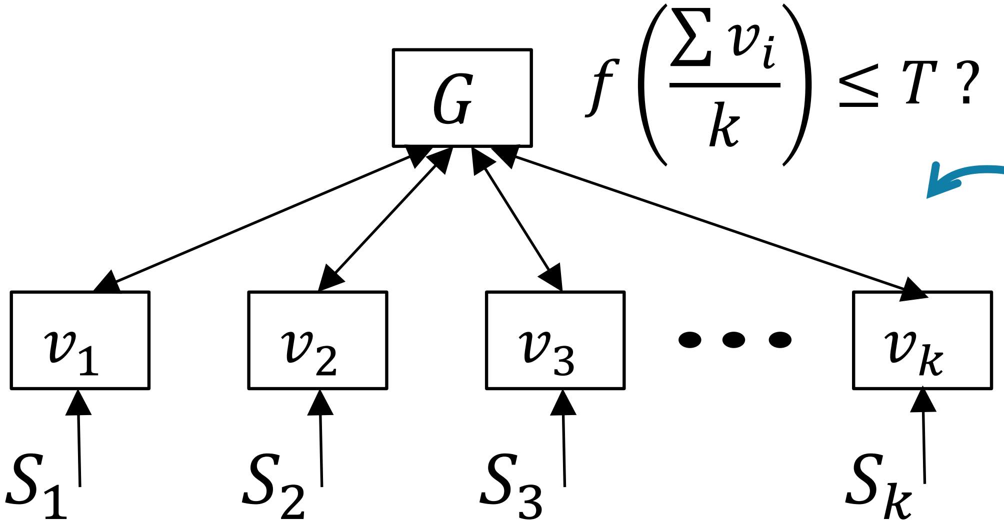
Monitoring over dynamic, distributed, streaming data

- Research initiated in 2003
- Supported by: ISF, BSF, Google, EC 7th Program ("*LIFT*" 2010-2013, "*FERARI*" 2014-2017, "SPEEDD" 2014-2017, "VaVEL" 2016-2019), others.

- Recent publications: SIGMOD12 **(dynamic case**), ICDE12 **(sensor networks),** TKDE12 **(shape sensitive),** VLDB13 **(sketches)**, ICDE14 **(skyline)**, NDSS14 **(privacy)**, TKDE14 **(heterogeneous case)**, IPDPS14 **(cloud health monitoring),** VLDB15 **(convex decomposition),** KDD15 **(Regression),** KDD16 **(convex bounds).**

Distributed Monitoring Model

- Distributed streams S_i continuously update the local statistics vectors v_i at the nodes
- The remote nodes communicate with a designated coordinator G
- The coordinator G must issue an alert when the global condition $f\left(\frac{\sum v_i}{k}\right) \leq T$ is breached



A rather general model, which describes many practically important problems. Can be further enriched by augmenting the local vectors v_i with functions of their raw coordinates.

Naïve monitoring

- When an update arrives at a node, the node calculates the statistics vector and updates the coordinator
- The coordinator has the true global state at all times, and it can check for threshold crossing
- This scheme suffers from huge
  - communication overhead
  - bandwidth requirement
  - computational load on the nodes and on the coordinator
  - energy overhead for computation and communication
- “less naïve”? Periodical update of coordinator leads to inherent tradeoff communication vs. latency

Minimizing communication - GM

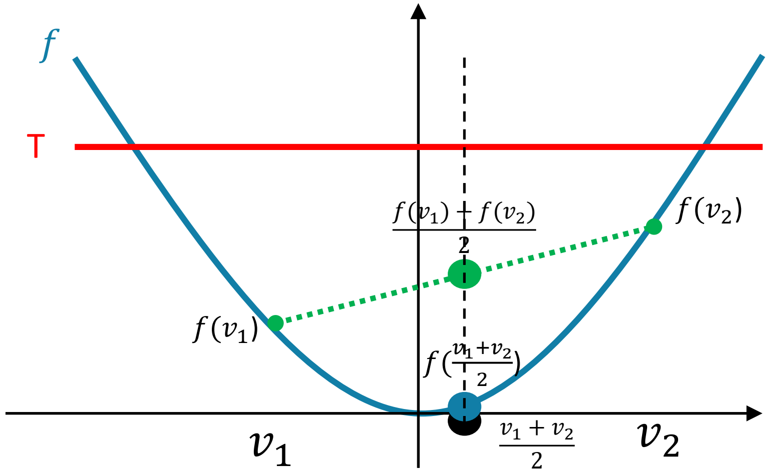
[SIGMOD 2016, Best Paper Honorary Mention]

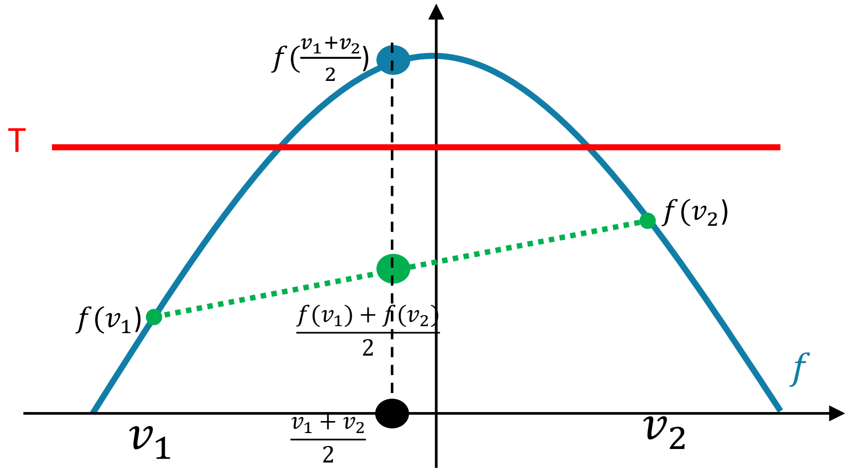
- We want to communicate only if the global condition might be breached
- The key is to decompose the global threshold condition into a set of local conditions on the statistics vectors at the nodes.
- As long as all local conditions are upheld no communication is required. “Quiescence”
- In case one of the conditions has been violated, the nodes need to communicate in order to resolve the violation.
- Great results but high computational complexity, problem for wireless environments

Monitoring convex functions [KDD16]

Reminder:
we wish to monitor $f\left(\frac{v_1+\cdots+v_n}{n}\right) \le T$

- If $f(v_i) \le$ T holds at every node, it also holds that $f\left(\frac{v_1+\cdots+v_n}{n}\right) \le T$
- Monitoring f (from above) is trivial – **simply monitor its value at every node**



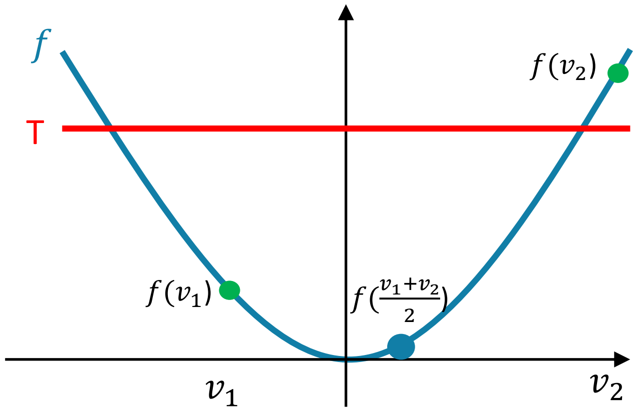


Left: The function is convex, so if $f(v_1) \le$ T and $f(v_2) \le$ T then $\frac{f(v_1)+f(v_2)}{2} \le T \Rightarrow f(\frac{v_1+v_2}{2}) \le T$

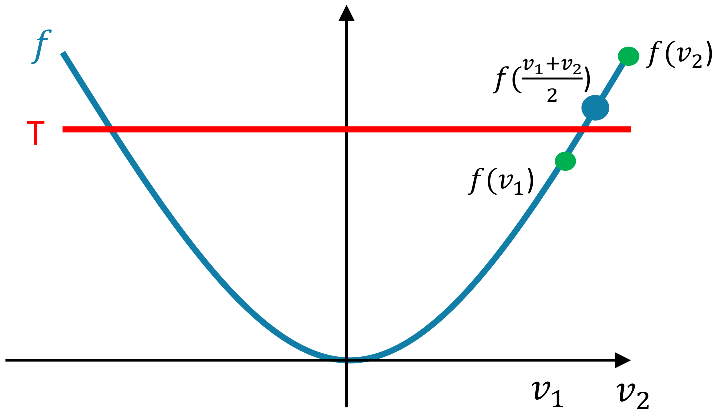
Right: The function is non-convex $f\left(\frac{v_1+v_2}{2}\right) > T$

Local Violations

- If the global threshold is crossed some local condition must be violated
- However a local condition may be violated, while the global condition holds



Local violation only. $f(v_2) > T$, but $f\left(\frac{v_1+v_2}{2}\right) \leq T$



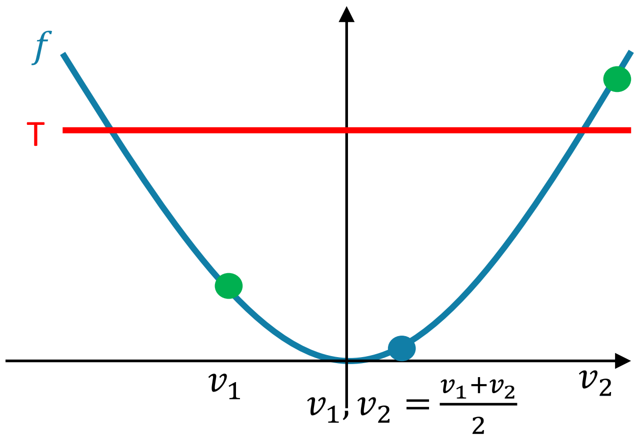
Local & Global violation. $f(v_2) > T$ and $f\left(\frac{v_1+v_2}{2}\right) > T$

Synchronization

- At t=0, the coordinator collects all the local vectors v_i
- It calculates the ref point – $\mathrm{p0} = \frac{v_1+\cdots+v_n}{n}$, and updates the nodes.
- Each node adjusts it's local vector to p0
- As updates arrive the vectors drift away from p0

At t=0 $f(v2) > T$, but $f\left(\frac{v_1+v_2}{2}\right) \leq T$

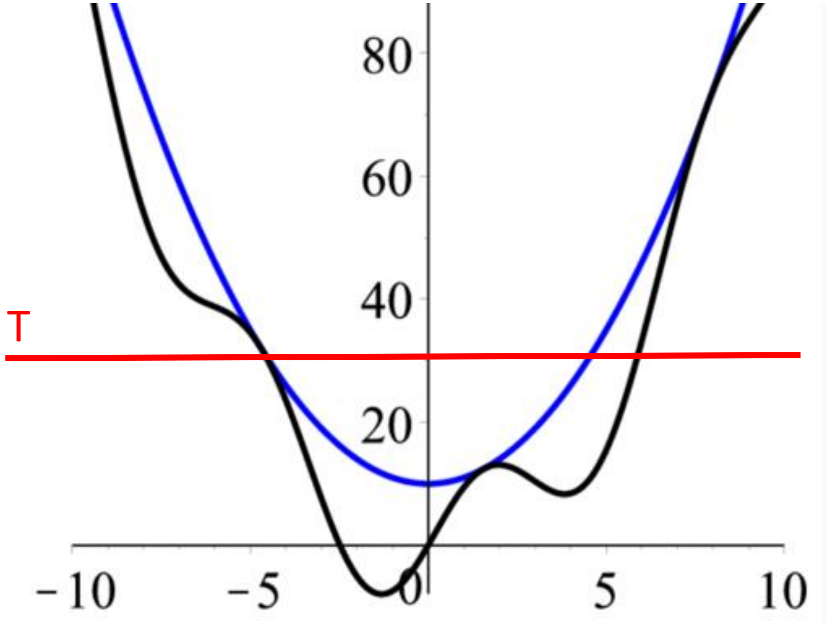
Synchronization:
Calculate $\mathrm{p0} = \frac{v_1+v_2}{2}$, and broadcast to nodes.
$v_1 = v_2 = p0 \Rightarrow f(v_1) = f(v_2) \leq T.$



Reminder:
we wish to monitor
$f\left(\frac{v_1+\cdots+v_n}{n}\right) \leq T$

Monitoring non-convex functions

- Find a convex function c such that c(u) ≥ f(u)
- Monitor the condition c ≤ T.
- While c ≤ T also f ≤ T



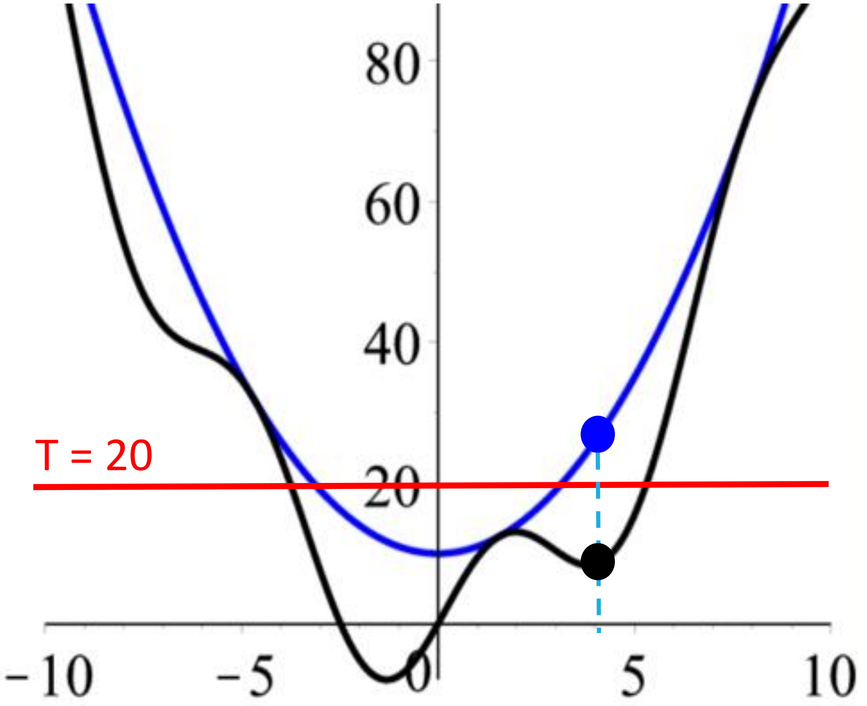
$f(x) = x^2 + 10\sin(x)$
$c(x) = x^2 + 10$

The Gap

- We replaced $f \leq T$ by $c \leq T$ where c is convex and $f \prec$ c

- Careful! False alarms. Can be eliminated
  - At the price of high-complexity computation
  - Hopefully, the gap is small and false alarms are rare

False alarm!
For x = 4, we get c(4) > T, so we must issue an alert, but f(4) ≤ T

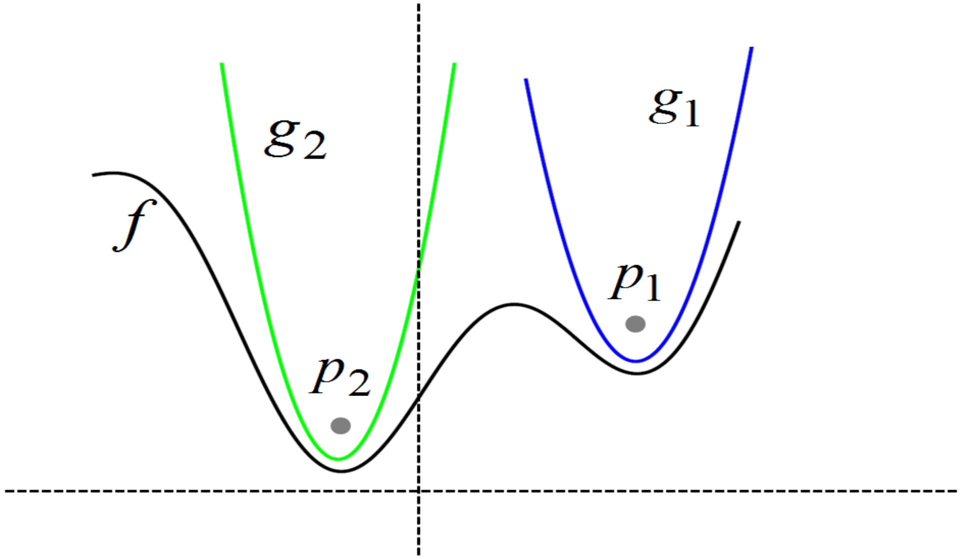


$f(x) = x^2 + 10\sin(x)$
$c(x) = x^2 + 10$

A “good bound”?

- It is impossible to choose a single global optimal c.
- The selection of the best tight bound depends on the ref-point
- … and on the future drift of the data streams

Both g_1 and g_2 are bounds of f.
The best bound depends on the ref point.
Clearly, g_1 is better around p_1, while g_2 is better around p_2.



Choosing a good bound

To gurantee correctness the function c -

- must be convex
- must bound f from above

To avoid false alarms – c should

- "Stick" to the monitored function as much as possible

To reduce communication – c must

- Leave large margins for the actual average to drift away from the ref point

"Convexizing" threshold conditions

Lemma.

If f possesses bounded second derivatives in a domain D, it can be expressed as the difference of two convex functions.

Proof.

Since the elements of H_f are bounded over D, there is an upper bound, B, on the absolute values of H_f 's negative eigenvalues.

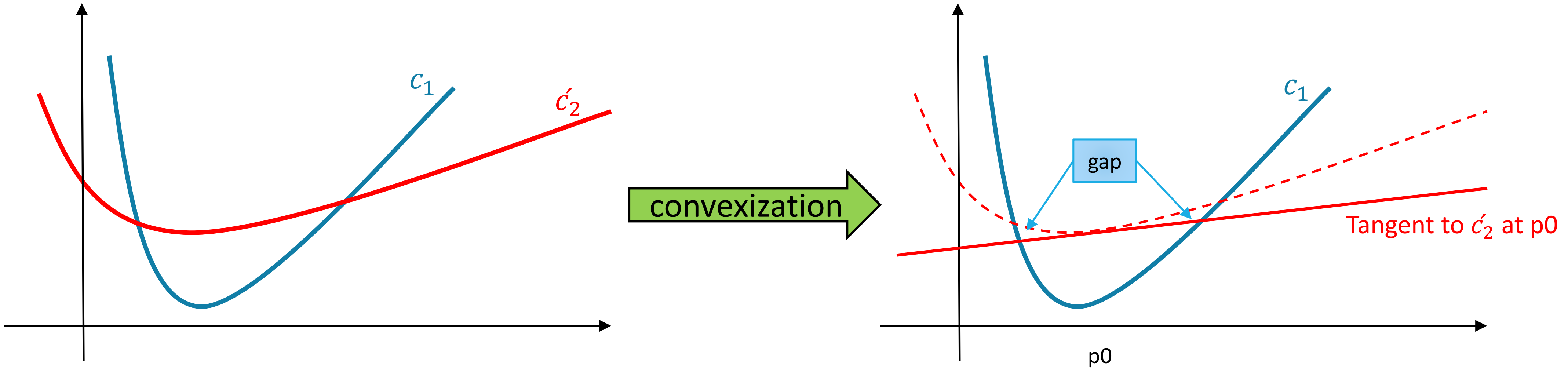
Define: $c_1(u) = f(u) + \frac{\boldsymbol{B}}{2}||u||^2, c_2(u) = \frac{\boldsymbol{B}}{2}||u||^2$

Clearly $f = c_1 - c_2$ and c_2 is positive definite. Also, $H_{c_1} = H_f + H_{c_2} = H_f + BI$

Hence all the eigenvalues of H_{c_1} are ≥ 0 and c_1 is convex.

"Convexizing" threshold conditions

- Note that $f = c_1 - c_2 \leq T \Leftrightarrow c_1 \leq T + c_2$
- The condition can be expressed as $c_1 \leq \acute{c}_2$ (where $\acute{c}_2 = T + c_2$)
  - Notice that both c_1 and $\acute{c}_2$ are convex
- This condition is "convexization friendly" – replace $\acute{c}_2$ with its tangent at p0

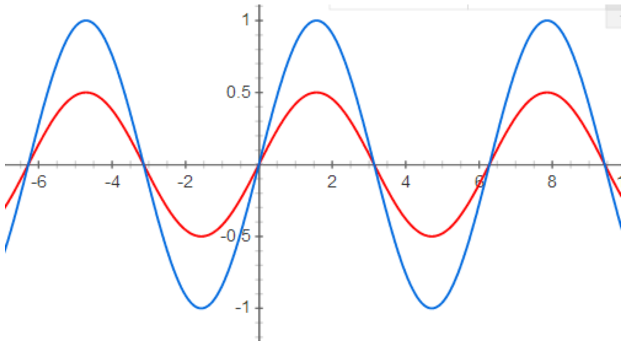


Application

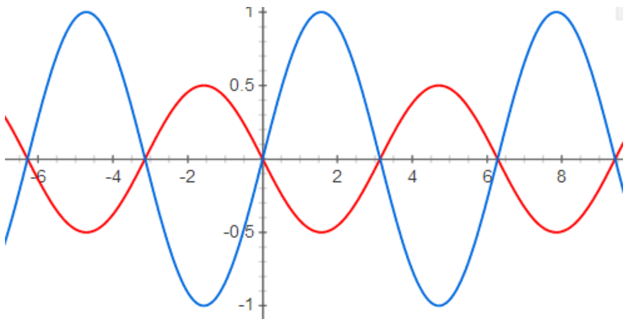
- We used CB (Convex Bound) to monitor four popular functions:
  - Inner product
  - Pearson correlation coefficient
  - Cosine similarity
  - PCA Score
- These functions have great practical importance
- They have no simple, efficient monitoring solutions
  - These functions are not linear, convex, concave, or monotonic

Pearson correlation coefficient

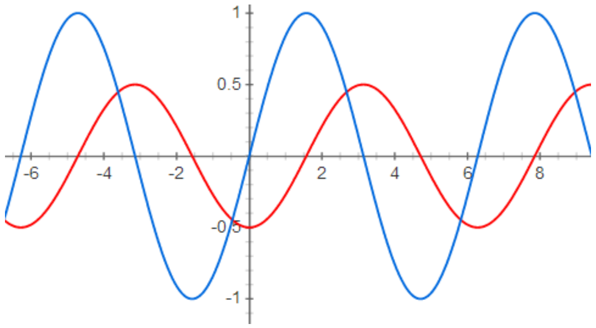
- PCC gives a value between -1 and 1
- 1 - total positive correlation. Whenever X appears Y also appears.
- 0 - no correlation. X and Y are totally independent, appearance of X says nothing about Y
- -1 - total negative correlation. Whenever X appears Y does not appear.



Positive correlation



Negative correlation



No correlation

Convexizing Pearson

- Assume T >0 , the condition $\frac{z-xy}{\sqrt{(x-x^2)(y-y^2)}} \leq T$ can be written as :

$$\underbrace{z - T\sqrt{(x-x^2)(y-y^2)}}_{\text{convex}} - xy \leq 0$$

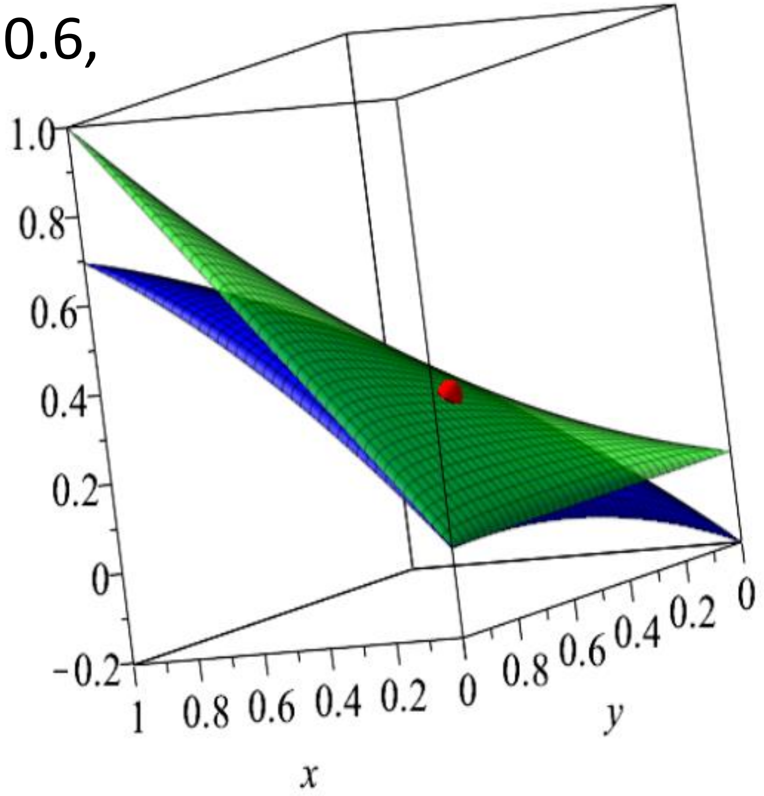
- xy is neither convex nor concave, so we use $xy = \underbrace{\frac{(x+y)^2}{4}}_{\text{Q1 (convex)}}$ - $\underbrace{\frac{(x-y)^2}{4}}_{\text{Q2 (convex)}}$, to get:

$$\underbrace{z - T\sqrt{(x-x^2)(y-y^2)} + Q2}_{\text{convex}} - \underbrace{Q1}_{\text{convex}} \leq 0$$

- This is difference between two convex functions – easy to convexize

Convexizing Pearson

A concave lower bound (blue) for PCC (green).
The reference point (in red) is x0 = 0.3, y0 = 0.6,
and T = 0.4.



Evaluation

Data: three real-life data sets:

- Reuters Corpus (RCV1-v2) - processed by Lewis et al
  - 804,414 categorized news documents , 47,236 features
- Twitter crawl (Dataset-UDI-TwitterCrawlAug2012) - by Li et al.
- KDD Cup 1999

Worst case runtime for a single check

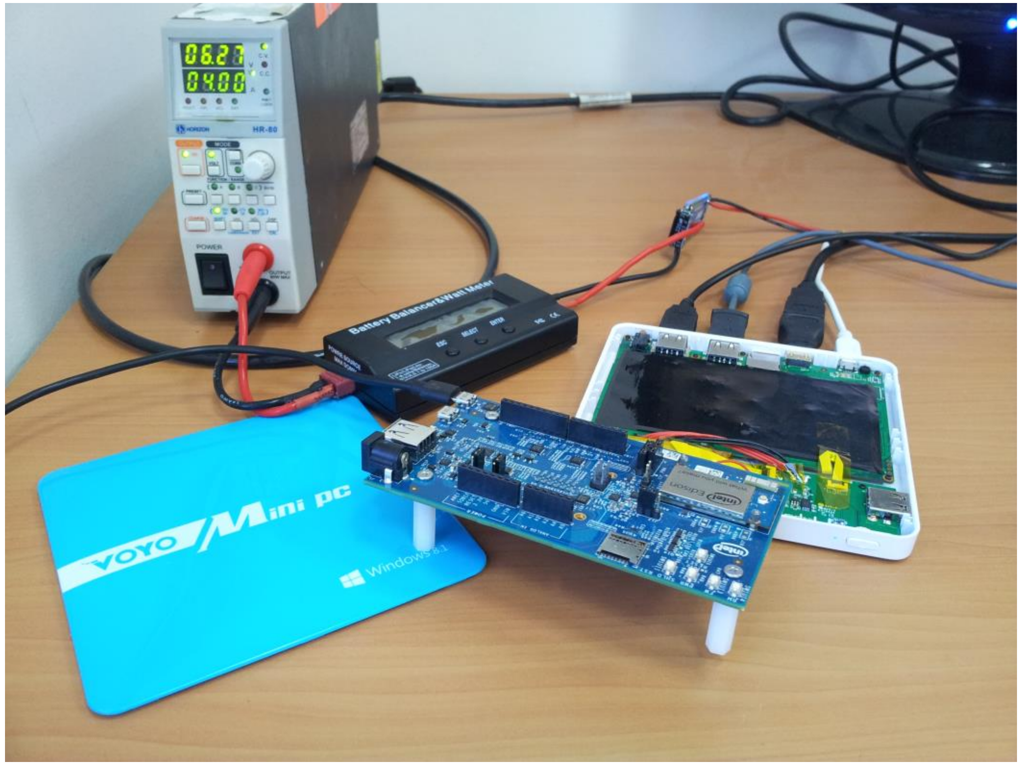
Function	Runtime (milliseconds)		Speedup
	GM	CB	
PCC	580	0.067	$8x10^3$
Inner-prod	1.82	0.089	20
CSIM	170,000	0.167	10^6

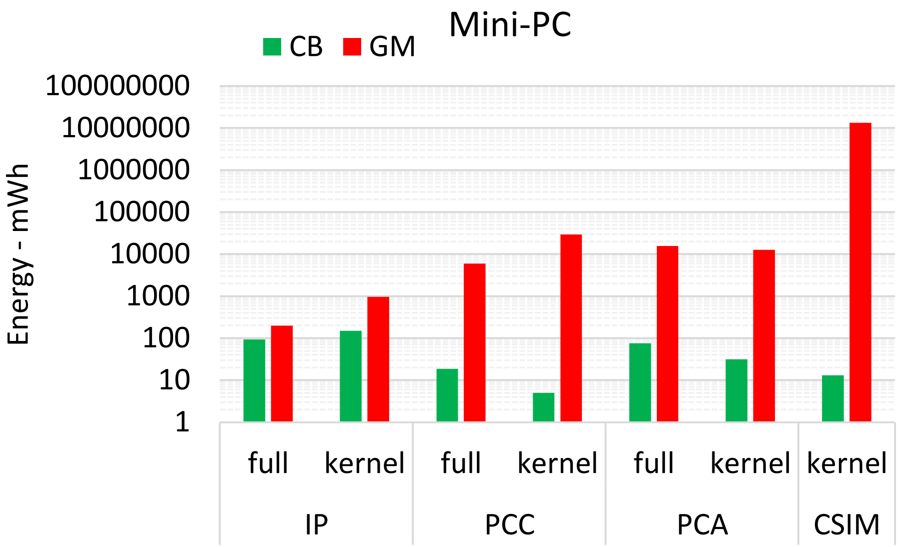
Over 3 days!

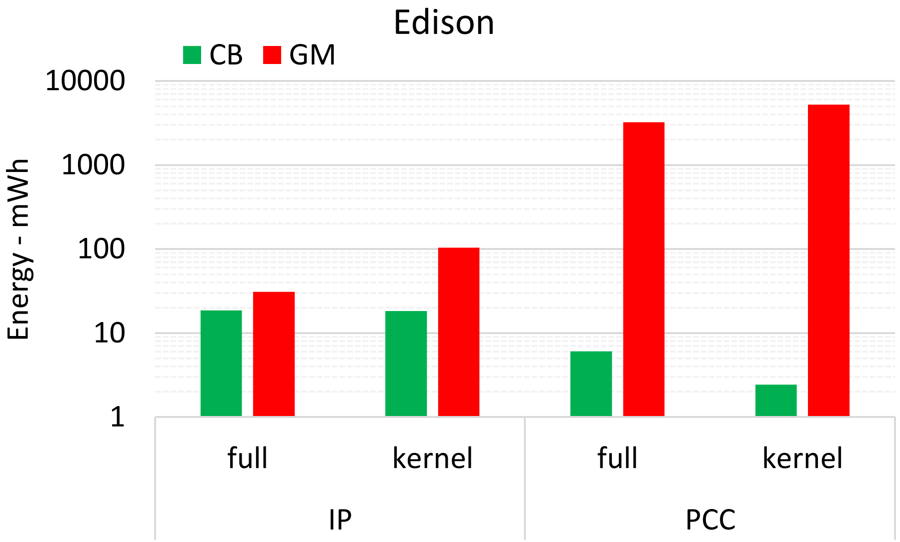
Overall runtime for monitoring the entire stream

Function	Runtime (minutes)		Speedup
	GM	CB	
PCC	4740	4	10^3
Inner-prod	200	68	3
CSIM	NA	130	NA

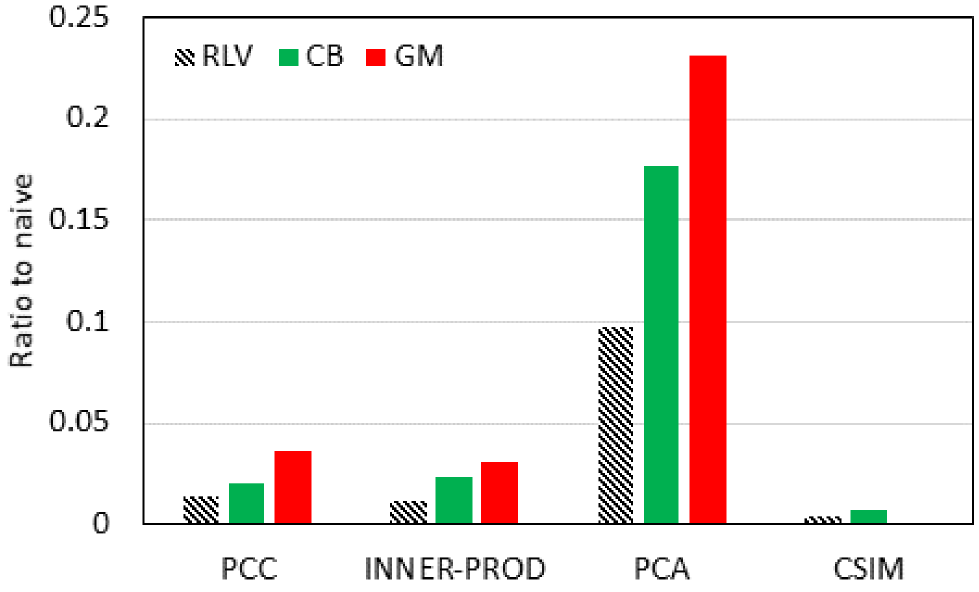
Power Consumption



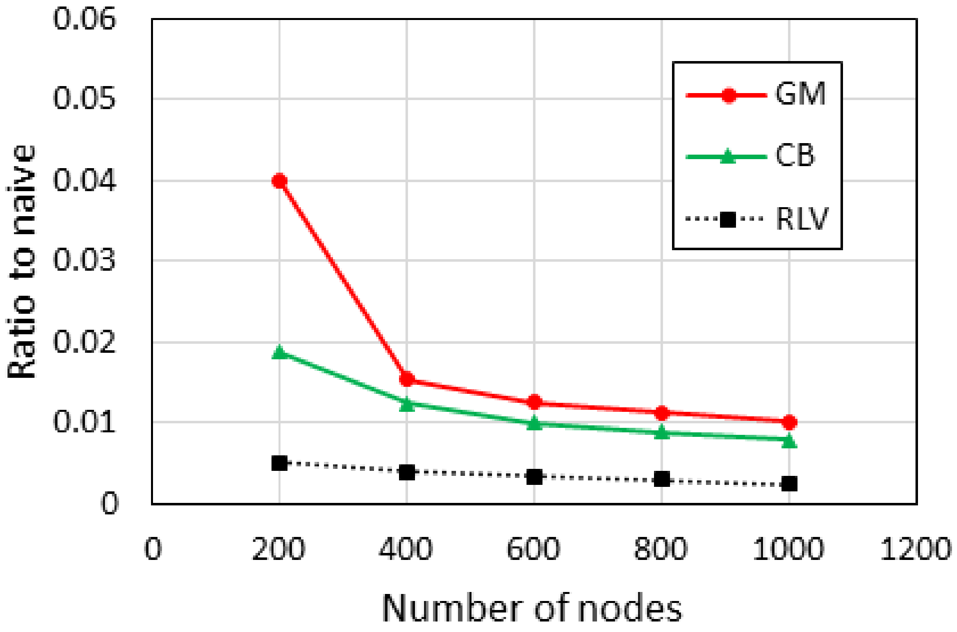




Communication Reduction



Communication ratio to the naive for CB and GM, and a super-optimal lower bound RLV. Each bar represents results across multiple thresholds and datasets. CB is always better than GM and very close to the lower bound. Csim lacks results for GM as the experiments did not complete in over 24 hours

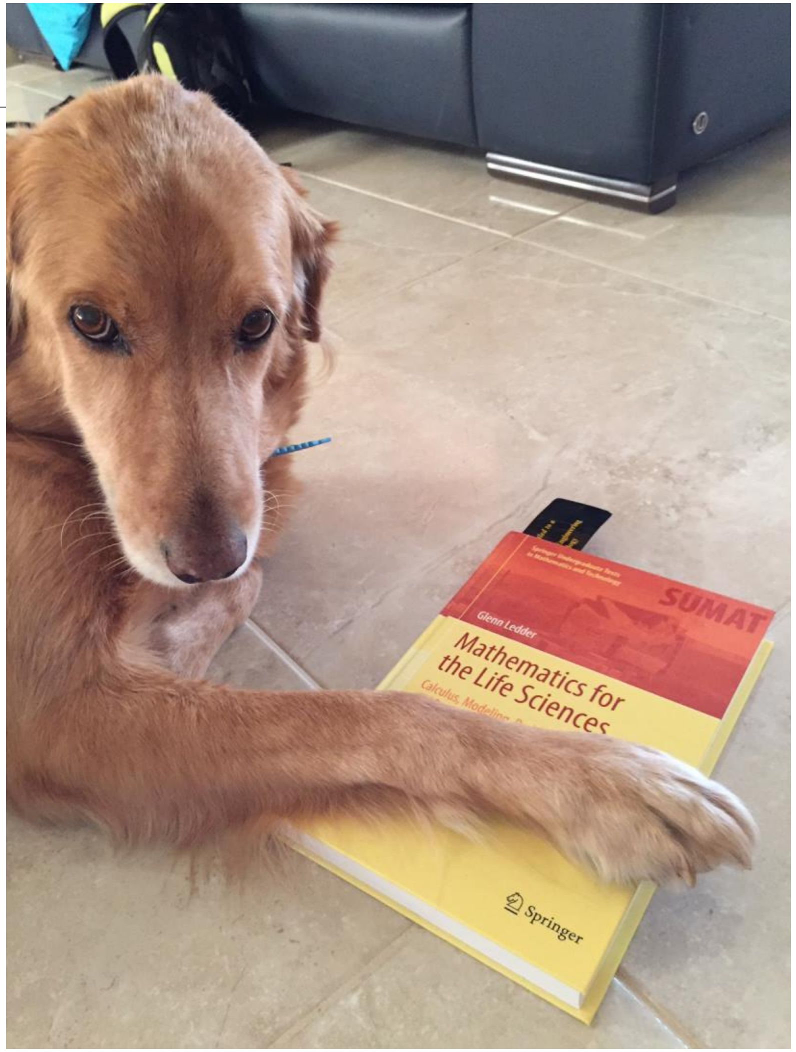


communication reduction results for the Inner-prod function on TWIT, using 200 to 1000 nodes.
All methods improve as the number of nodes grow. CB remains closer to the RLV bound and maintains its advantage over GM.

Summary

- A new method for monitoring threshold functions over distributed streams
- Runtime lower by orders of magnitude compared to state of the art
- Reduced communication overhead

- Future work:
  - Further applications
  - Alternative methods to “convexize” monitoring problems
  - Implementation on smart systems, IoT




Thank you