Continuous Distributed Monitoring

A Short Survey

Graham Cormode

AT&T Labs

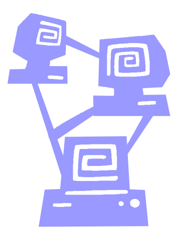
Distributed Monitoring

There are many scenarios where we need to track events:

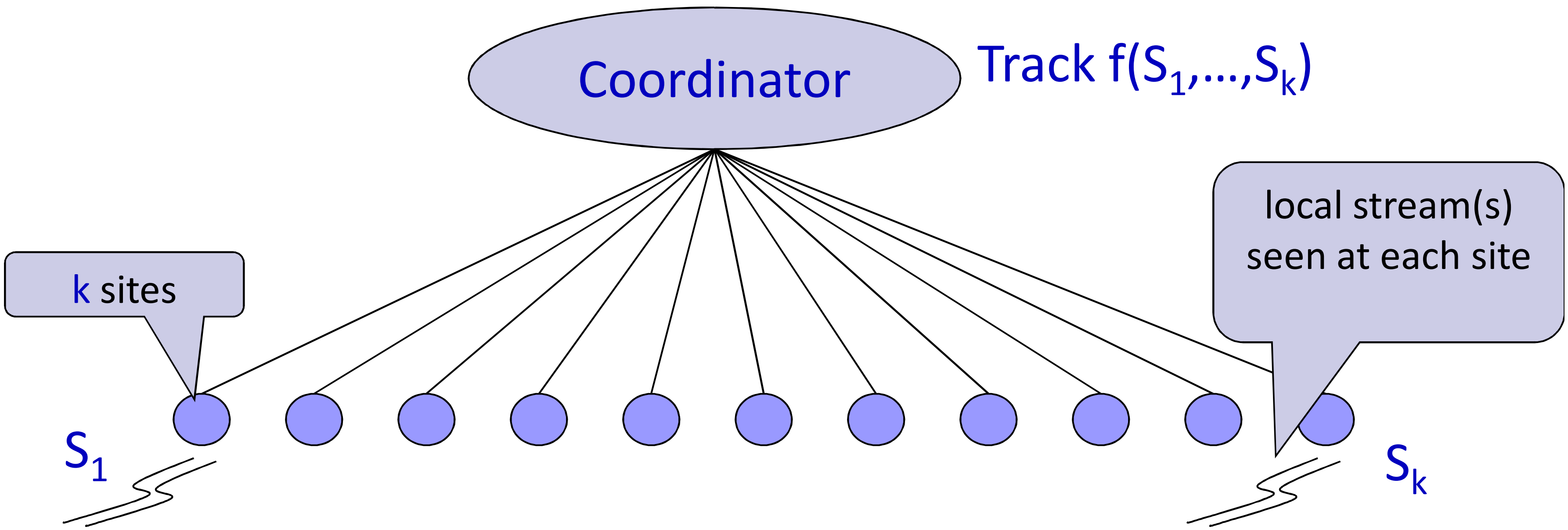
- Network health monitoring within a large ISP
- Collecting and monitoring environmental data with sensors
- Observing usage and abuse of distributed data centers

All can be abstracted as a collection of observers who want to collaborate to compute a function of their observations

From this we generate the Continuous Distributed Model



Continuous Distributed Model



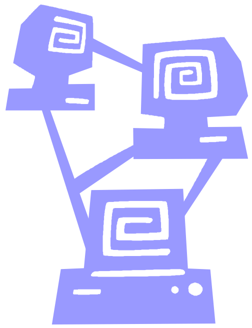
- Site-site communication only changes things by factor 2
- **Goal**: Coordinator *continuously tracks* (global) function of streams
  - Achieve communication poly(k,1/ε,log n)
  - Also bound space used by each site, time to process each update

Challenges

- Monitoring is Continuous…
  - Real-time tracking, rather than one-shot query/response
- …Distributed…
  - Each remote site only observes part of the global stream(s)
  - Communication constraints*:* must minimize monitoring burden
- …Streaming…
  - Each site sees a high-speed local data stream and can be resource (CPU/memory) constrained
- …Holistic…
  - Challenge is to monitor the complete global data distribution
  - Simple aggregates (e.g., aggregate traffic) are easier

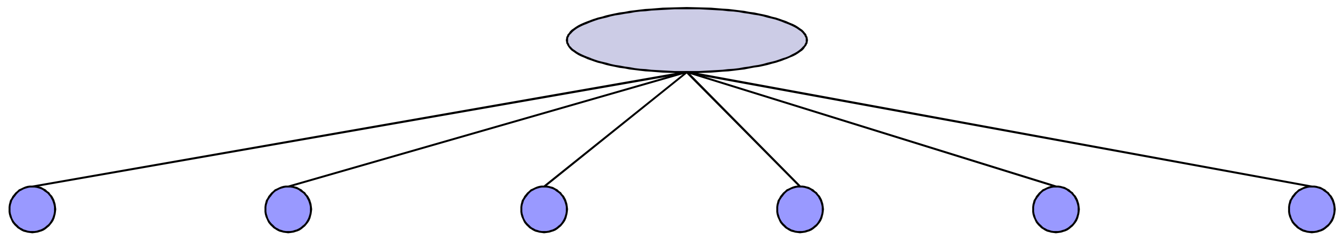
Baseline Approach

- Sometimes periodic polling suffices for simple tasks
  - E.g., SNMP polls total traffic at coarse granularity
- Still need to deal with holistic nature of aggregates
- Must balance polling frequency against communication
  - Very frequent polling causes high communication, excess battery use in sensor networks
  - Infrequent polling means delays in observing events
- Need techniques to reduce communication while guaranteeing rapid response to events



Variations in the model

- Multiple streams define the input A
- Given function f, several types of problem to study:
  - Threshold Monitoring: identify when $f(A) > \tau$
    Possibly tolerate some approximation based on $\varepsilon\tau$
  - Value Monitoring: always report accurate approximation of $f(A)$
  - Set Monitoring: $f(A)$ is a set, always provide a "close" set
- Direct communication between sites and the coordinator
  - Other network structures possible (e.g., hierarchical)

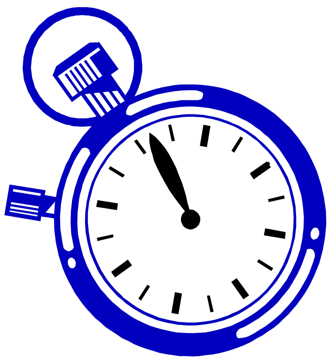


Outline

1. The Continuous Distributed Model
2. **How to count to 10**
3. Entropy, a non-linear function
4. The geometric approach
5. A sample of sampling
6. Prior work and future directions

The Countdown Problem

- A first abstract problem that has many applications
- Each observer sees events
- Want to alert when a total of τ events have been seen
  - Report when more than 10,000 vehicles have passed sensors
  - Identify the 1,000,000th customer at a chain of stores
- Trivial solution: send 1 bit for each event, coordinator counts
  - $O(\tau)$ communication
  - Can we do better?



A First Approach

- One of k sites must see τ/k events before threshold is met
- So each site counts events, sends message when τ/k are seen
- Coordinator collects current count n_i from each site
  - Compute new threshold $\tau' = \tau - \sum_{i=1}^{k} n_i$
  - Repeat procedure for τ' until $\tau' < k$, then count all events
- Analysis: $\tau > \tau'/(1-1/k) > \tau''/(1-1/k)^2 > \ldots$
  - Number of thresholds $= \log (\tau/k) / \log(1/(1-1/k)) = O(k \log (\tau/k))$
  - Total communication: $O(k^2 \log (\tau/k))$ [each update costs $O(k)$]
- Can we do better?

A Quadratic Improvement

- Observation: $O(k)$ communication per update is wasteful
- Try to wait for more updates before collecting
- Protocol operates over $\log(\tau/k)$ rounds [C.,Muthukrishnan, Yi 08]
  - In round j, each site waits to receive $\tau/(2^j k)$ events
  - Subtract this amount from local count n_i, and alert coordinator
  - Coordinator awaits k messages in round j, then starts round $j+1$
  - Coordinator informs all sites at end of each round
- Analysis: k messages in each round, $\log(\tau/k)$ rounds
  - Total communication is $O(k \log(\tau/k))$
  - Correct, since total count can't exceed τ until final round

Approximate variation

- Sometimes, we can tolerate approximation
- Only need to know if threshold τ is reached approximately
- So we can allow some bounded uncertainty:
  - Do not report when count < $(1-\varepsilon)\ \tau$
  - Definitely report when count > τ
  - In between, do not care
- Previous protocol adapts immediately:
  - Just wait until distance to threshold reaches $\varepsilon\tau$
  - Cost of the protocol reduces to $O(k \log 1/\varepsilon)$ (independent of τ)

Extension: Randomized Solution

- Cost is high when k grows very large
- Randomization reduces this dependency, with parameter ε
- Now, each site waits to see $O(\varepsilon^2\tau/k)$ events
  - Roll a die: report with probability $1/k$, otherwise stay silent
  - Coordinator waits to receive $O(1/\varepsilon^2)$ reports, then terminates
- Analysis: in expectation, coordinator stops after $\tau(1-\varepsilon/2)$ events
  - With Chernoff bounds, show that it stops before τ events
  - And does not stop before $\tau(1-\varepsilon)$ events
- Gives a randomized, approximate solution: uncertainty of $\varepsilon\tau$

Outline

1. The Continuous Distributed Model
2. How to count to 10
3. **Entropy, a non-linear function**
4. The geometric approach
5. A sample of sampling
6. Prior work and future directions

Monitoring Entropy

- Countdown solutions relied on monotonicity and linearity
- Entropy is a function which is neither monotone or linear!
- Let f_i be the total number of occurrences of item i
- Let m be the total number of all items = $\sum_i f_i$
- This defines an empirical probability distribution:
  - Item i has empirical probability f_i/m
- We want to monitor the entropy of this distribution:

  $$H = \sum_i f_i/m \log (m/f_i)$$

  - Specifically, report whether $H > \tau$ or $H < (1-\varepsilon)\tau$

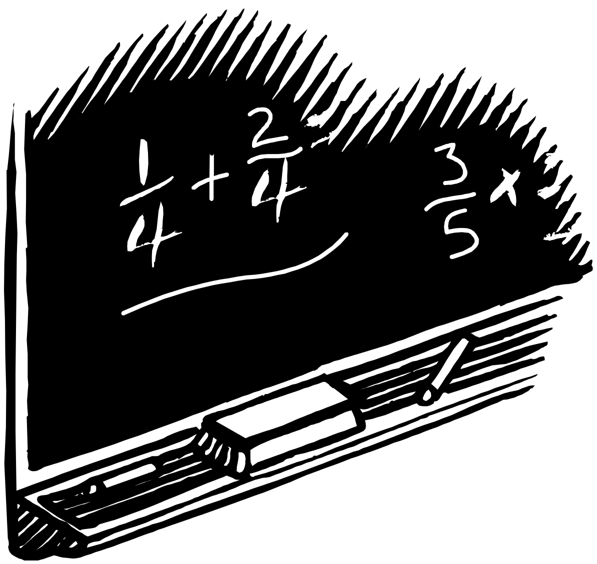
Entropy Protocol

- Protocol based on [Arackaparambil Brody Chakrabarti 09]
- Initially, collect all items from sites for 100 items (say)
  - Empirical entropy is changing rapidly here
- In each subsequent round i, coordinator computes τ_i
  - Run approximate countdown protocol for τ_i with $\varepsilon = ½$
  - Collect frequency distribution from all sites, compute entropy
- Analysis: suppose we have m items, and there are n arrivals
  - Can bound the change in entropy as $2n/(m+n) \log (m+n)$

Change in Entropy

- Entropy change as f_i goes to $(f_i + g_i)$ is at most

$$\sum_i \mid f_i / m \log (m/f_i) - (f_i + g_i)/(m+n) \log (m+n)/(f_i + g_i) \mid$$
$$\leq \sum_i \mid f_i/m \log (m+n) - (f_i + g_i)/(m+n) \log (m+n) \mid$$
$$\leq \sum_i \mid f_i / m - (f_i + g_i)/(m+n) \mid \log(m+n)$$
$$\leq \sum_i \mid f_i (m+n) - (f_i + g_i)m \mid \log (m+n) / m(m+n)$$
$$\leq \sum_i \mid f_i n - g_i m \mid \log (m+n)/m(m+n)$$
$$\leq \sum_i (f_i n + g_i m)/m(m+n) \log (m+n)$$
$$\leq (mn + mn)/m(m+n) \log (m+n)$$
$$\leq 2n/(m+n) \log (m+n)$$



Entropy Protocol Analysis

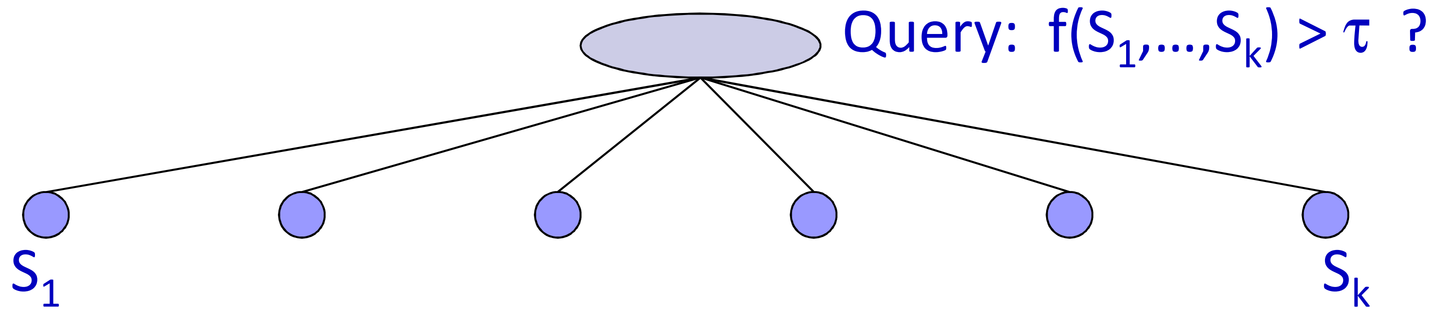
- Change in entropy is at most $2n/(m+n) \log (m+n)$
  - If we set $n < m$, then this is bounded by $2n/m \log (2m)$
- Need to know if entropy changes by at least $\varepsilon\tau/2$
  - (the smallest amount to force coordinator to change output)
- So set $\tau_i = \varepsilon\tau m/(4 \log 2m)$
  - So long as n is less than this, entropy changes by at most $\varepsilon\tau/2$
- Analysis: letting N be total number of observations so far,
  - Observations increase by a $(1+ \varepsilon\tau/4 \log 2N)$ factor each round
  - Bounds total number of rounds as $O((\log^2 N)/\varepsilon\tau)$
  - Countdown protocol costs $O(k)$ per round

Extension: Entropy Sketches

- Currently, each site sends current distribution each round
  - If there are D distinct items seen, total cost is $O(kD(\log^2 N)/(\varepsilon\tau))$
  - Can be very costly when D is high!
- Solution: send a compact sketch of the data distribution
  - Sketches for entropy give a $1\pm\varepsilon$ approximation in $O(1/\varepsilon^2)$ space
  - Sketches are combined to produce a sketch of the whole dbn
  - Total cost is $O(k/(\tau\varepsilon^3) \log^2 N)$
- Lower bound for deterministic algorithms: $\Omega(k\varepsilon^{-1/2} \log (\varepsilon N/k))$
  - Room for improvement in dependence on ε, $\log N$

Outline

General Non-linear Functions



- For general, non-linear f(), the problem becomes a lot harder!
  - E.g., information gain over global data distribution
- Non-trivial to decompose the global threshold into “safe” local site constraints
  - E.g., consider $N=(N_1+N_2)/2$ and $f(N) = 6N - N^2 > 1$
    Tricky to break into thresholds for $f(N_1)$ and $f(N_2)$

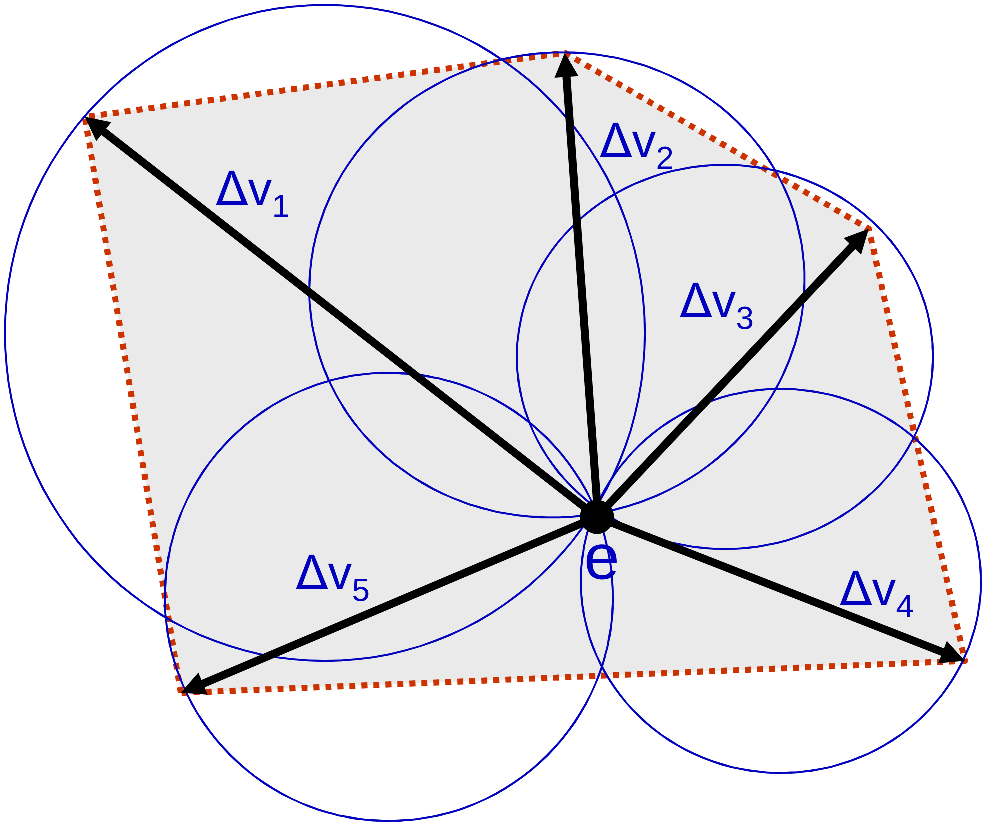
The Geometric Approach

- A general purpose geometric approach [Scharfman et al.'06]
- Each site tracks a local statistics *vector* v_i (e.g., data distribution)
- Global condition is $f(v) > \tau$, where $v = \sum_i \lambda_i v_i$ $(\sum_i \lambda_i = 1)$
  - v = convex combination of local statistics vectors
- All sites share estimate $e = \sum_\iota \lambda_i v_i'$ of v

  based on latest update v_i' from site i
- Each site i tracks its drift from its most recent update $\Delta v_i = v_i - v_i'$

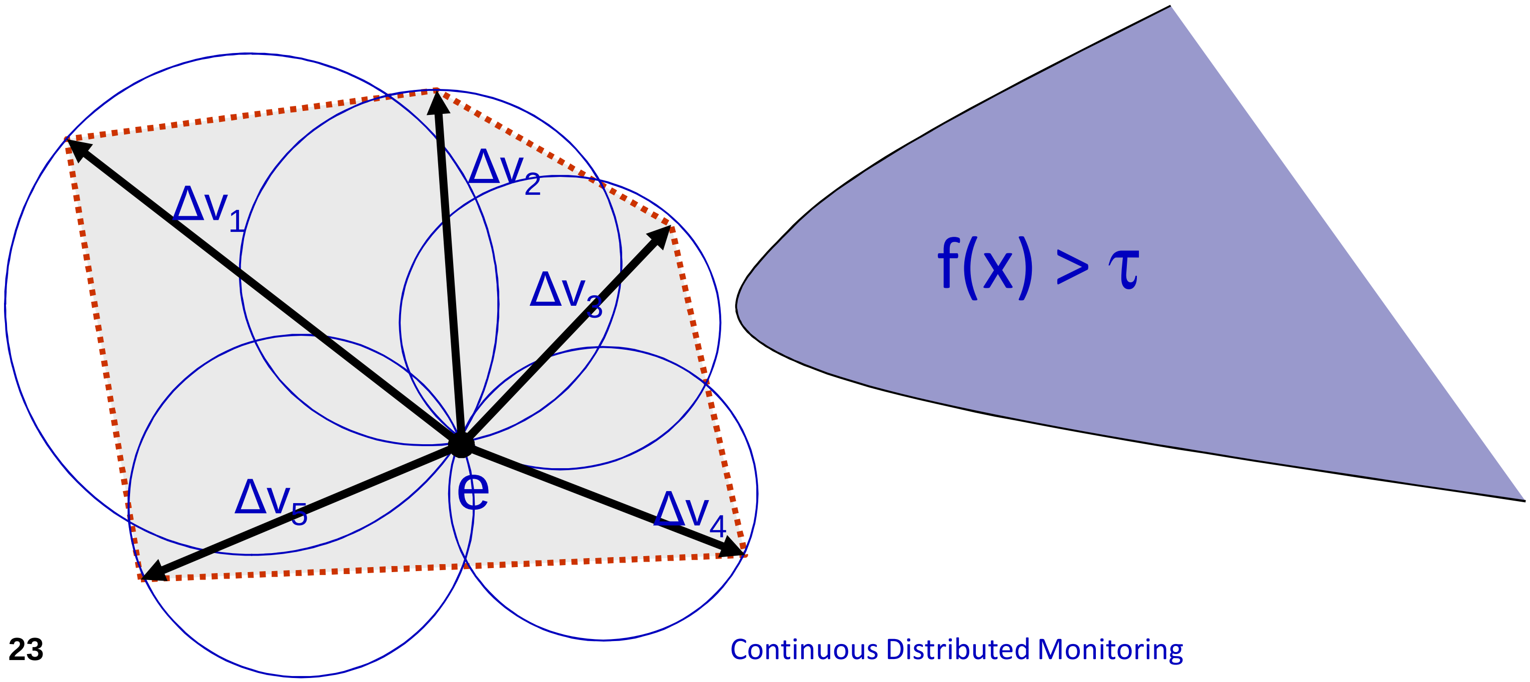
Covering the convex hull

- Key observation: $v = \sum_i \lambda_i \cdot (e+\Delta v_i)$
  (a convex combination of "translated" local drifts)
- v lies in the convex hull of the $(e+\Delta v_i)$ vectors
- Convex hull is completely covered by spheres with radii $||\Delta v_i/2||_2$ centered at $e+\Delta v_i/2$
- Each such sphere can be constructed independently



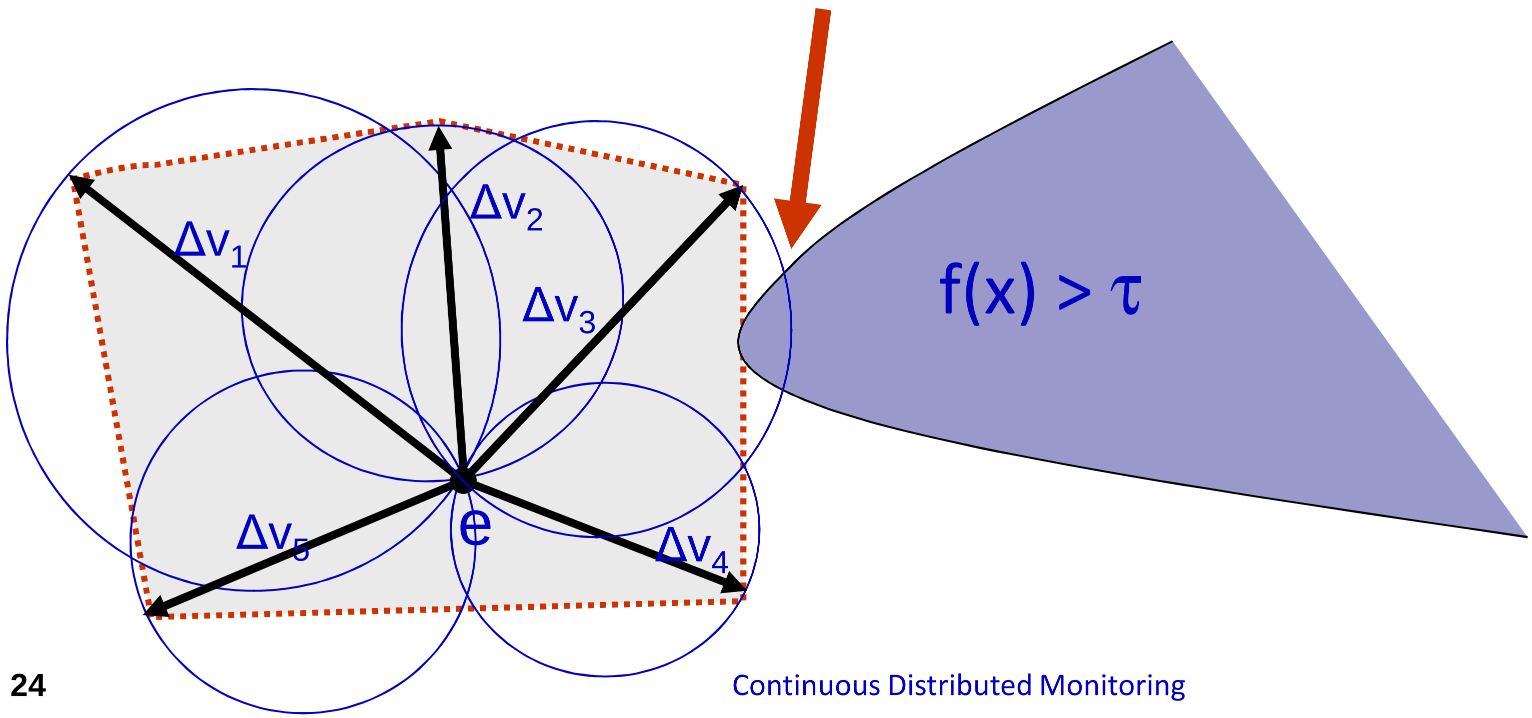
Monochromatic Regions

- Monochromatic Region: For all points x in the region $f(x)$ is on the same side of the threshold ($f(x) > \tau$ or $f(x) \leq \tau$)
- Each site independently checks its sphere is monochromatic
  - Find max and min for f() in local sphere region (may be costly)
  - Broadcast updated value of v_i if not monochrome



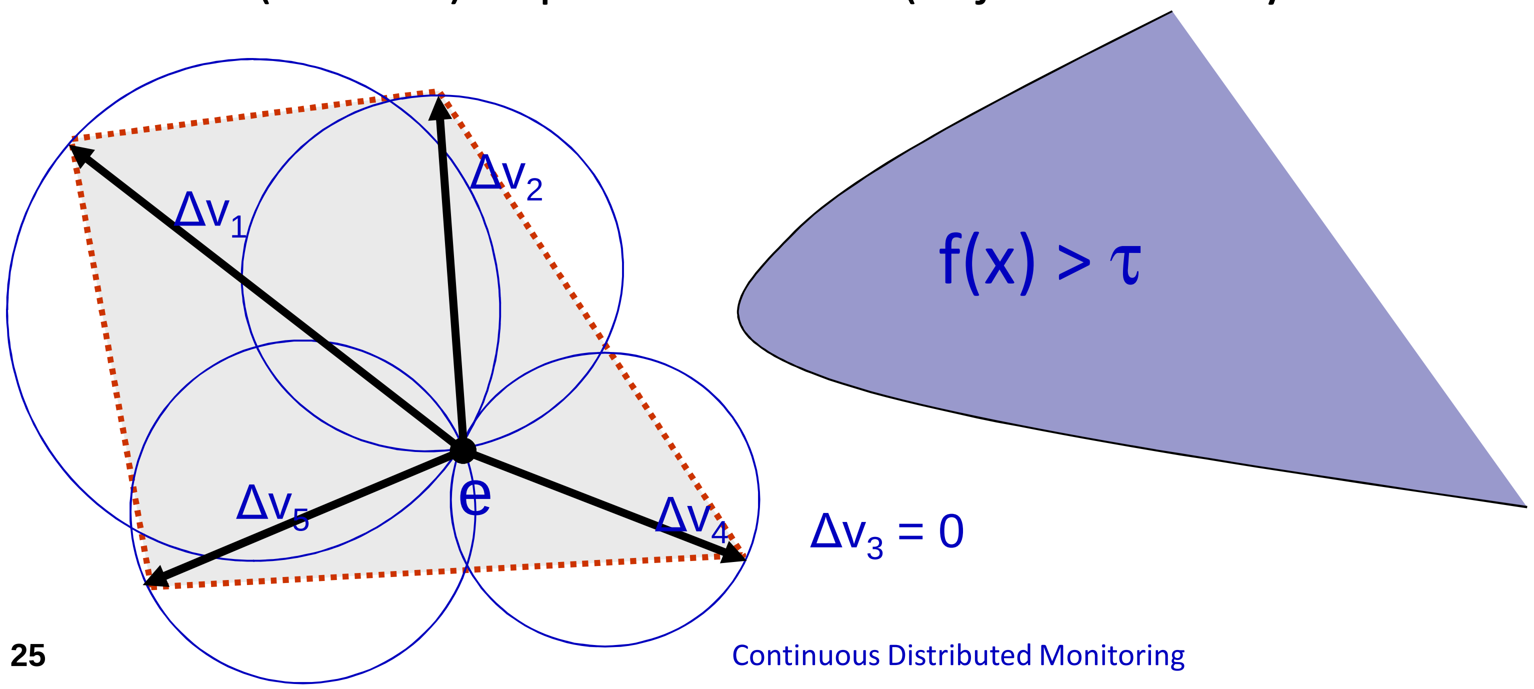
Restoring Monochomicity

- After broadcast, $||\Delta v_i||_2 = 0 \Rightarrow$ Sphere at i is monochromatic



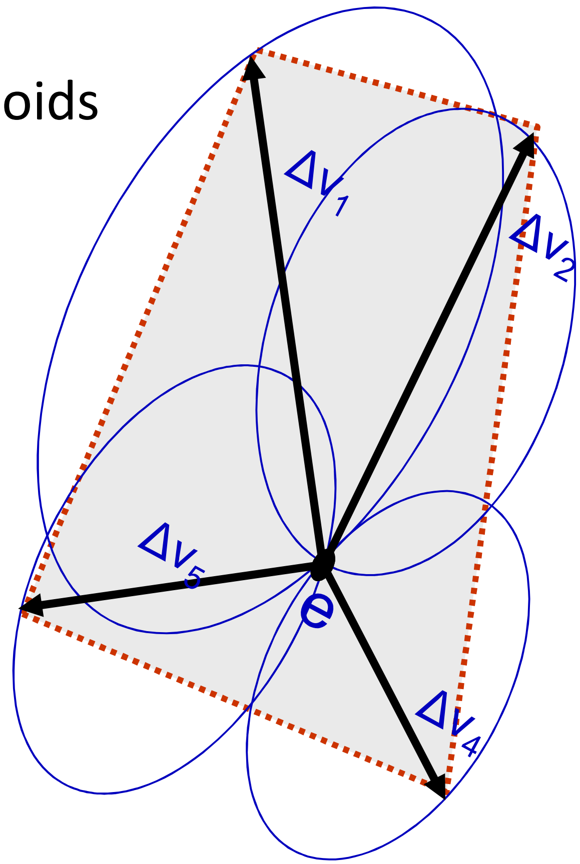
Restoring Monochomicity

- After broadcast, $||\Delta v_i||_2 = 0 \Rightarrow$ Sphere at i is monochromatic
  - Global estimate e is updated, which may cause more site update broadcasts
- Coordinator case: Can allocate local slack vectors to sites to enable "localized" resolutions
  - Drift (=radius) depends on slack (adjusted locally for subsets)



Extension: Transforms and Shifts

- Subsequent extensions further reduce cost [Scharfman et al. 10]
  - Same analysis of correctness holds when spheres are allowed to be ellipsoids
  - Additional offset vectors can be used to increase radius when close to threshold values
  - Combining these observations allows additional cost savings



Outline

1. The Continuous Distributed Model
2. How to count to 10
3. Entropy, a non-linear function
4. The geometric approach
5. **A sample of sampling**
6. Prior work and future directions

Drawing a Sample

- A basic 'set monitoring' problem is to draw a uniform sample
- Given inputs of total size N, draw a sample of size s
  - Uniform over all subsets of size s
- Overall approach:
  - Define a general sampling technique amenable to distribution
  - Bound the cost
  - Extend to sliding windows

Binary Bernoulli Sampling

- Always sample with probability $p = 2^{-i}$
- Randomly pick i bits, each of which is 0/1 with probability ½
- Select item if all i random bits are 0
- (Conceptually) store the random bits for each item
  - Can easily pick more random bits if the sampling rate decreases

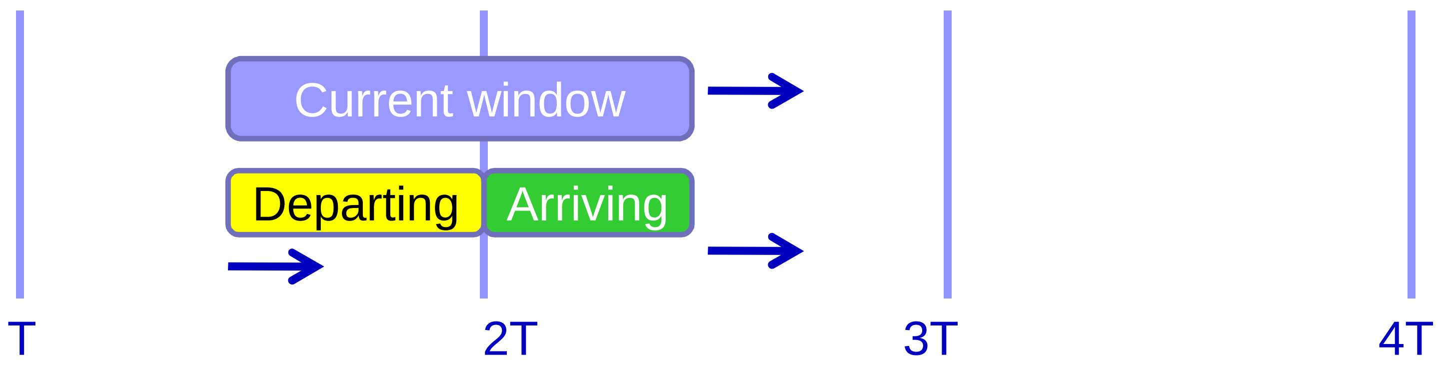
Sampling Protocol

- Protocol based on [C., Muthukrishnan, Yi, Zhang 10]
- In round i, each site samples with $p = 2^{-i}$
  - Sampled items are sent to the coordinator
  - Coordinator picks one more random bit
  - End round i when coordinator has s items with (i+1) zeros
  - Coordinator informs each site that a new round has started
  - Coordinator picks extra random bits for items in its sample

Protocol Costs

- Correctness: coordinator always has (at least) s items
  - Sampled with the same probability p
  - Can subsample to reach exactly s items
- Cost: each round is expected to send $O(s)$ items total
  - Can bound this with high probability via Chernoff bounds
  - Number of rounds is similar bounded as $O(\log N)$
  - Communication cost is $O((k+s) \log N)$
- Lower bound on communication cost of $\Omega(k + s \log N)$
  - At least this many items are expected to appear in the sample
  - $O(k \log (k/sN) + s \log n)$ upper bound by adjusting probabilities

Extension: Sliding Window



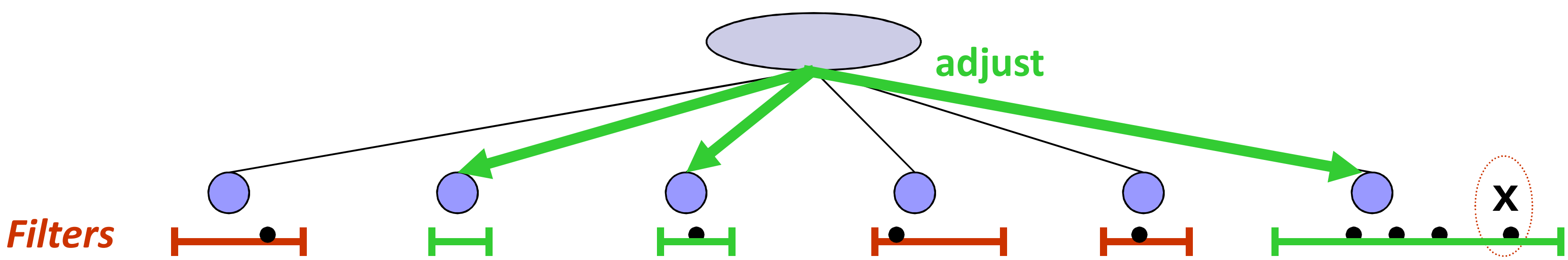
- Extend to sliding windows: only sample from last T arrivals
- Key insight: can break window into 'arriving' and 'departing'
  - Use multiple instances of Countdown protocol to track expiries
- Cost of such a protocol is $O(ks \log (W/s))$
  - Near-matching $\Omega(ks \log(W/ks))$ lower bound

Outline

1. The Continuous Distributed Model
2. How to count to 10
3. Entropy, a non-linear function
4. The geometric approach
5. A sample of sampling
6. **Prior work and future directions**

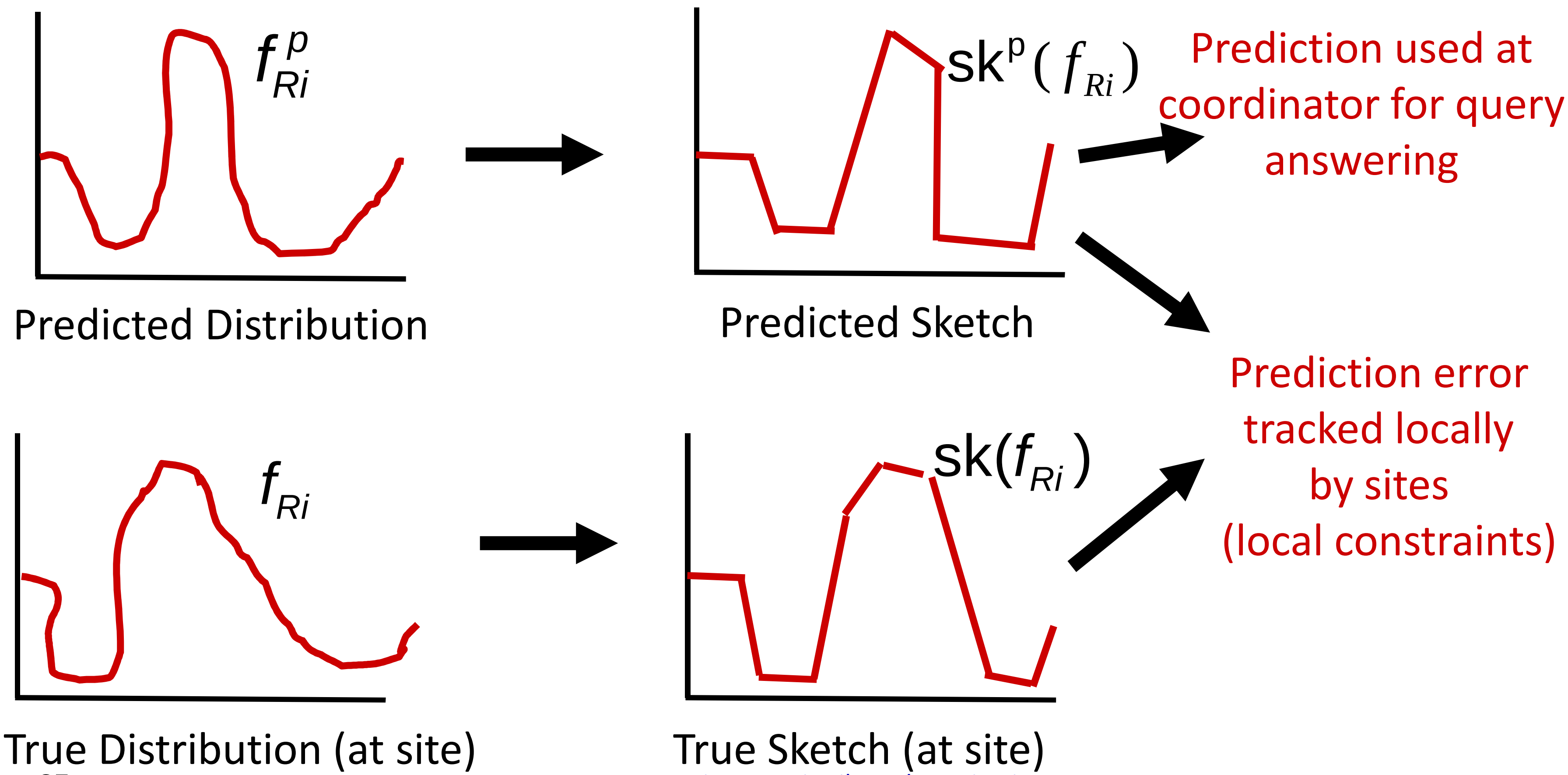
Early Work

- Continuous distributed monitoring arose in several places:
  - Networks: Reactive monitoring [Dilman Raz 01]
  - Databases: Distributed triggers [Jain et al. 04]
- Initial work on tracking multiple values
  - "Adaptive Filters" [Olston Jiang Widom 03]
  - Distributed top-k [Babcock Olston 03]



Prediction Models

- Prediction further reduces cost [C, Garofalakis, Muthukrishnan, Rastogi 05]
  - Combined with approximate (sketch) representations

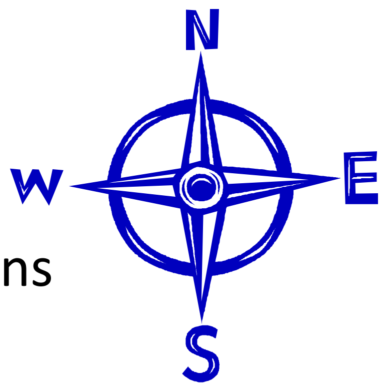


Problems in Distributed Monitoring

- Much interest in these problems in TCS and Database areas
- Many specific functions of (global) data distribution studied:
  - Set expressions [Das Ganguly Garofalakis Rastogi 04]
  - Quantiles and heavy hitters [C, Garofalakis, Muthukrishnan, Rastogi 05]
  - Number of distinct elements [C., Muthukrishnan, Zhuang 06]
  - Conditional Entropy [Arackaparambil, Bratus, Brody, Shubina 10]
  - Spectral properties of data matrix [Huang et al. 06]
  - Anomaly detection in networks [Huang et al. 07]
- Track functions only over sliding window of recent events
  - Samples [C, Muthukrishnan, Yi, Zhang 10]
  - Counts and frequencies [Chan Lam Lee Ting 10]

Other Work

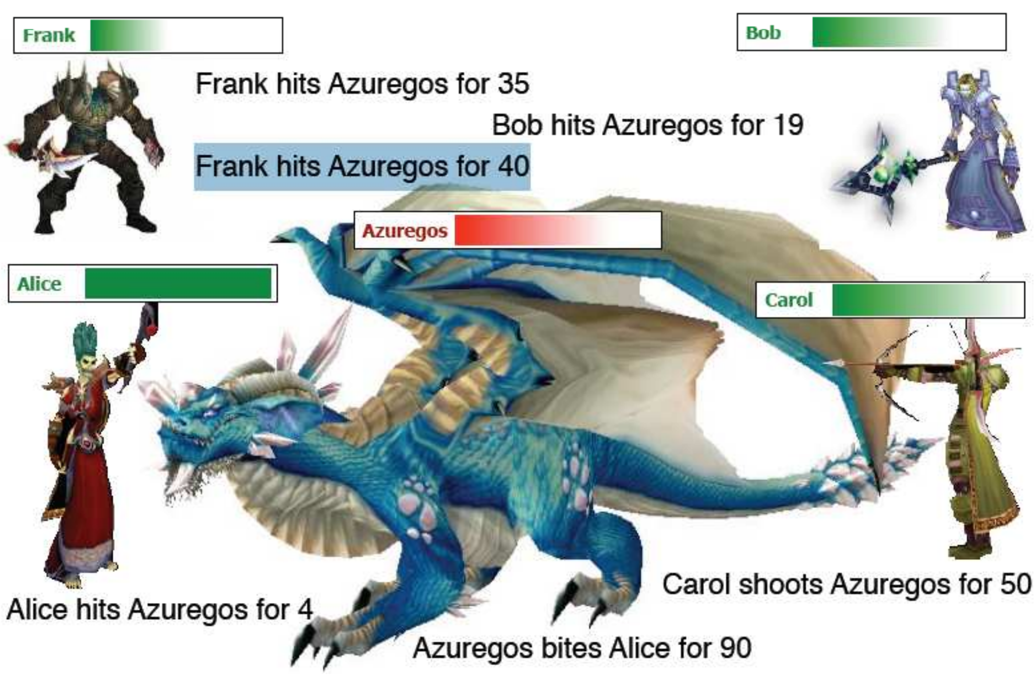
- Many open problems remain in this area
  - Improve bounds for previously studied problems
  - Provide bounds for other important problems
  - Give general schemes for larger classes of functions



- Much ongoing work
  - See EU-support LIFT project, `lift-eu.org`
- Two specific open problems:
  - Develop systems and tools for continuous distributed monitoring
  - Provide a deeper theory for continuous distributed monitoring

Monitoring Systems

- Much theory developed, but less progress on deployment
- Some empirical study in the lab, with recorded data
- Still applications abound: Online Games [Heffner, Malecha 09]
  - Need to monitor many varying stats and bound communication

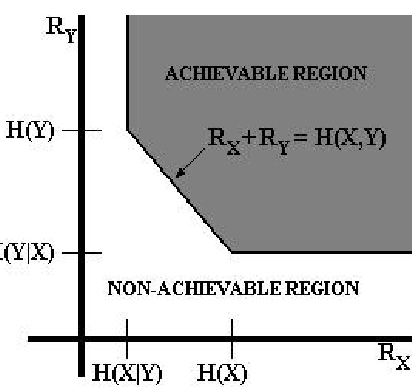


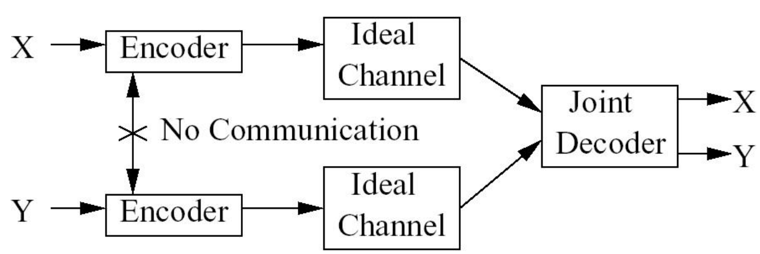
- Severa
  - Buil ems
  - Evol buted DBMSs?)
- Severa
  - Wha specific?
  - Wha onitoring?

Theoretical Foundations

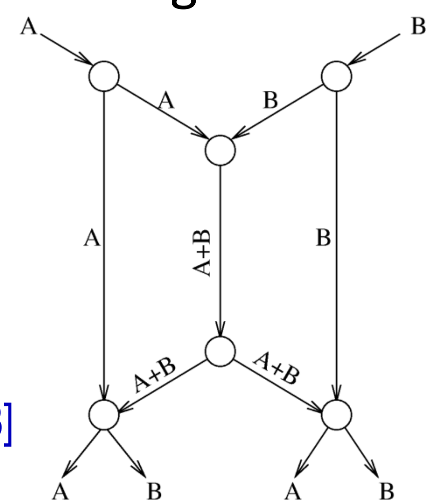
"Communication complexity" studies lower bounds of distributed one-shot computations

- Gives lower bounds for various problems, e.g., **count distinct** (via reduction to abstract problems)
- Need new theory for continuous computations
  - Based on info. theory and models of how streams evolve?
  - Link to distributed source coding or network coding?





Slepian-Wolf theorem [Slepian Wolf 1973]



http://www.networkcoding.info/

https://buffy.eecs.berkeley.edu/PHP/resabs/resabs.php?f_year=2005&f_submit=chapgrp&f_chapter=1

Concluding Remarks

- Continuous distributed monitoring is a natural model
- Captures many real world applications
- Much non-trivial work in this model
- Much work remains to do!

Thank You!

References (1)

[Babcock, Olston 03] B. Babcock and C. Olston. Distributed top-k monitoring. In ACM SIGMOD Intl. Conf. Management of Data, 2003.

[Chan Lam Lee Ting 10] H.-L. Chan, T.-W. Lam, L.-K. Lee, and H.-F. Ting. Continuous monitoring of distributed data streams over a time-based sliding window. In Symp. Theoretical Aspects of Computer Science, 2010.

[Cormode, Garofalakis '05] G. Cormode and M. Garofalakis. Sketching streams through the net: Distributed approximate query tracking. In Proceedings of the International Conference on Very Large Data Bases, 2005.

[Cormode Garofalakis, Muthukrishnan Rastogi 05] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In Proceedings of ACM SIGMOD International Conference on Management of Data, 2005.

[C., Muthukrishnan, Zhuang 06] G. Cormode, S. Muthukrishnan, and W. Zhuang. What's different: Distributed, continuous monitoring of duplicate resilient aggregates on data streams. In IEEE Intl. Conf. Data Engineering, 2006.

[Cormode, Muthukrishnan, Yi 08] G. Cormode, S. Muthukrishnan, and K. Yi. Algorithms for distributed, functional monitoring. In ACM-SIAM Symp. Discrete Algorithms, 2008.

References (2)

[Cormode, Muthukrishnan, Yi, Zhang, 10] G. Cormode, S. Muthukrishnan, K. Yi, and Q. Zhang. Optimal sampling from distributed streams. In ACM Principles of Database Systems, 2010.

[Das Ganguly Garofalakis Rastogi 04] A. Das, S. Ganguly, M. Garofalakis, and R. Rastogi. Distributed Set-Expression Cardinality Estimation. In Proceedings of VLDB, 2004.

[Dilman, Raz 01] M. Dilman, D. Raz. Efficient Reactive Monitoring. In IEEE Infocom, 2001.

[Heffner, Malecha 09] K. Heffner and G. Malecha. Design and implementation of generalized functional monitoring. www.people.fas.harvard.edu/~gmalecha/proj/funkymon.pdf, 2009.

[Huang et al. 06] L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A. Joseph, and N. Taft. Distributed PCA and Network Anomaly Detection. In NIPS, 2006.

[Huang et al. 07] L. Huang, M. N. Garofalakis, A. D. Joseph, and N. Taft. Communication-efficient tracking of distributed cumulative triggers. In ICDCS, 2007.

[Jain et al. 04] A. Jain, J.M.Hellerstein, S. Ratnasamy, D. Wetherall. A Wakeup Call for Internet Monitoring Systems: The Case for Distributed Triggers. In Proceedings of HotNets-III, 2004.

[Kerlapura et al. 06] R. Kerlapura, G. Cormode, and J. Ramamirtham. Communication-efficient distributed monitoring of thresholded counts. In ACM SIGMOD, 2006.

References (3)

[Olston, Jiang, Widom 03] C. Olston, J. Jiang, J. Widom. Adaptive Filters for Continuous Queries over Distributed Data Streams. In ACM SIGMOD, 2003.

[Sharfman et al. 06] I. Sharfman, A. Schuster, D. Keren: A geometric approach to monitoring threshold functions over distributed data streams. SIGMOD Conference 2006: 301-312

[Sharfman et al. 10] I. Sharfman, A. Schuster, and D. Keren. Shape-sensitive geometric monitoring. In ACM Principles of Database Systems, 2010.

[Slepian, Wolf 73] D. Slepian, J. Wolf. Noiseless coding of correlated information sources. IEEE Transactions on Information Theory, 19(4):471-480, July 1973.