Efficient Threshold Monitoring for Distributed Probabilistic Data

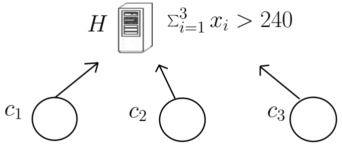
Mingwang Tang, Feifei Li, Jeff M. Phillips, Jeffrey Jestes

THE UNIVERSITY OF UTAH®

School of Computing, University of Utah

Outline

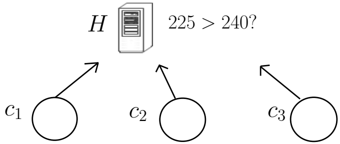
- Distributed threshold monitoring(DTM) problem:



t_1	70	80	75
t_2	70	90	90
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_T	70	80	70

Introduction

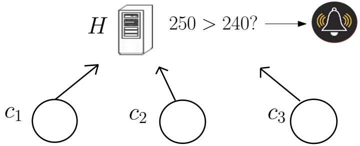
- Distributed threshold monitoring(DTM) problem:



t_1	70	80	75
t_2	70	90	90
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_T	70	80	70

Introduction

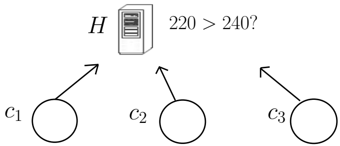
- Distributed threshold monitoring(DTM) problem:



t_1	70	80	75
t_2	70	90	90
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_T	70	80	70

Introduction

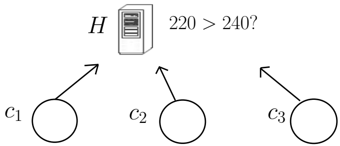
- Distributed threshold monitoring(DTM) problem:



t_1	70	80	75
t_2	70	90	90
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_T	70	80	70

Introduction

- Distributed threshold monitoring(DTM) problem:



t_1	70	80	75
t_2	70	90	90
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_T	70	80	70

- Extensively studied: e.g., S. Jeyashanker et al. propose an adaptive technique dealing with DTM problem ($\sum_{i=1}^{g} x_i \leq T$) for deterministic data.
- [ICDE08] S. Jeyashanker et al., Efficient Constraint Monitoring Using Adaptive Thresholds, ICDE 2008

- Distributed threshold monitoring(DTM) problem:

H  $\Sigma_{i=1}^{3} x_i > 240$

	c_1 B_1	c_2 B_2	c_3 B_3
t_1	70	80	75
t_2	70	90	90
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_T	70	80	70

- Extensively studied: e.g., S. Jeyashanker et al. propose an adaptive technique dealing with DTM problem ($\sum_{i=1}^{g} x_i \leq T$) for deterministic data.
- [ICDE08] S. Jeyashanker et al., Efficient Constraint Monitoring Using Adaptive Thresholds, ICDE 2008

- Distributed threshold monitoring(DTM) problem:

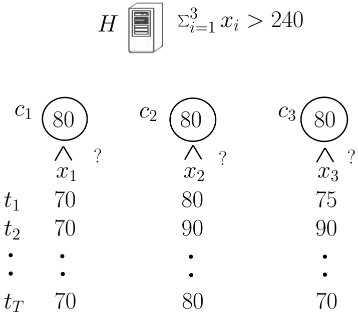
H  $\Sigma_{i=1}^{3} x_i > 240$

c_1 80 c_2 80 c_3 80

t_1	70	80	75
t_2	70	90	90
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_T	70	80	70

- Extensively studied: e.g., S. Jeyashanker et al. propose an adaptive technique dealing with DTM problem ($\sum_{i=1}^{g} x_i \leq T$) for deterministic data.
- [ICDE08] S. Jeyashanker et al., Efficient Constraint Monitoring Using Adaptive Thresholds, ICDE 2008

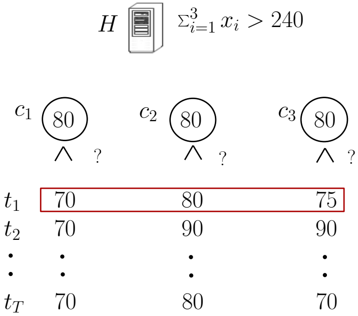
- Distributed threshold monitoring(DTM) problem:



- Extensively studied: e.g., S. Jeyashanker et al. propose an adaptive technique dealing with DTM problem ($\sum_{i=1}^{g} x_i \leq T$) for deterministic data.

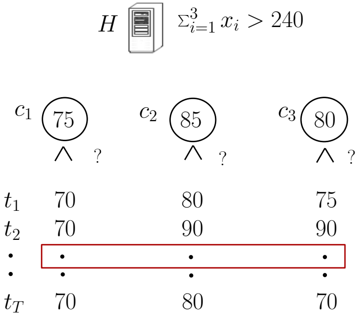
- [ICDE08] S. Jeyashanker et al., Efficient Constraint Monitoring Using Adaptive Thresholds, ICDE 2008

- Distributed threshold monitoring(DTM) problem:



- Extensively studied: e.g., S. Jeyashanker et al. propose an adaptive technique dealing with DTM problem ($\sum_{i=1}^{g} x_i \leq T$) for deterministic data.
- [ICDE08] S. Jeyashanker et al., Efficient Constraint Monitoring Using Adaptive Thresholds, ICDE 2008

- Distributed threshold monitoring(DTM) problem:



- Extensively studied: e.g., S. Jeyashanker et al. propose an adaptive technique dealing with DTM problem ($\sum_{i=1}^{g} x_i \leq T$) for deterministic data.

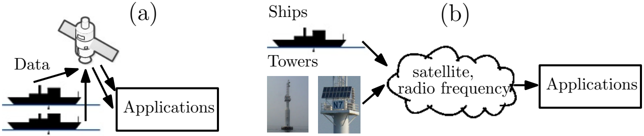
- [ICDE08] S. Jeyashanker et al., Efficient Constraint Monitoring Using Adaptive Thresholds, ICDE 2008

Introduction

- New challenge in the DTM problem: uncertainty naturally exist in distributed data
  - data integration produces fuzzy matches
  - noisy sensor readings

Introduction

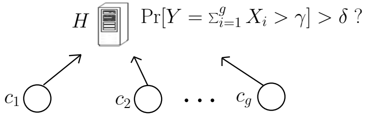
- New challenge in the DTM problem: uncertainty naturally exist in distributed data
  - data integration produces fuzzy matches
  - noisy sensor readings
- The Shipboard Automated Meteorological and Oceanographic System(SAMOS)



- Attribute-level uncertain model (with a single attribute score)

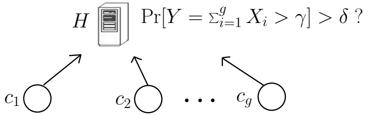
tuples	attribute score
d_1	$X_1 = \{(v_{1,1}, p_{1,1}), (v_{1,2}, p_{1,2})...(v_{1,b_1}, p_{1,b_1})\}$
d_2	$X_2 = \{(v_{2,1}, p_{2,1}), (v_{2,2}, p_{2,2})...(v_{2,b_2}, p_{2,b_2})\}$
.	...
d_t	$X_t = \{(v_{t,1}, p_{t,1}), (v_{t,2}, p_{t,2})...(v_{t,b_t}, p_{t,b_t})\}$

- Distributed probabilistic threshold monitoring (DPTM):



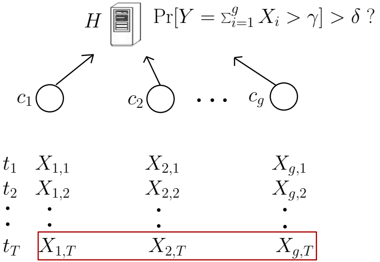
t_1	$X_{1,1}$	$X_{2,1}$	$X_{g,1}$
t_2	$X_{1,2}$	$X_{2,2}$	$X_{g,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_T	$X_{1,T}$	$X_{2,T}$	$X_{g,T}$

- Distributed probabilistic threshold monitoring (DPTM):

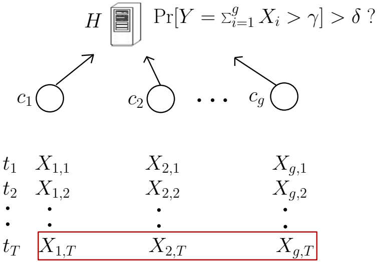


$$
\begin{array}{cccc}
t_1 & \boxed{X_{1,1} \quad X_{2,1} \quad X_{g,1}} \\
t_2 & X_{1,2} \quad X_{2,2} \quad X_{g,2} \\
\vdots & \vdots \quad \vdots \quad \vdots \\
t_T & X_{1,T} \quad X_{2,T} \quad X_{g,T}
\end{array}
$$

- Distributed probabilistic threshold monitoring (DPTM):



- Distributed probabilistic threshold monitoring (DPTM):



- Naive Method:
  - c_i sends X_i to H at each time instance t;
  - H computes $\Pr[Y > \gamma]$ based on X_i's
  - expensive in terms of both communication ($O(gT)$) and computation ($O(n^g T)$).

Our Approach

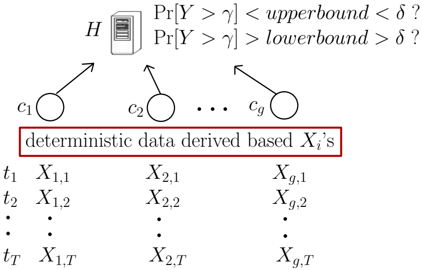
- Exact Methods:
  - Computing $\Pr[Y > \gamma]$ exactly is expensive

Our Approach

- Exact Methods:
  - Computing $\Pr[Y > \gamma]$ exactly is expensive
  - Incorporates pruning techniques.

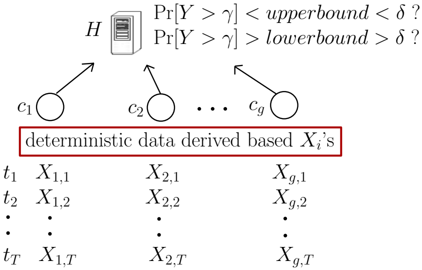
Our Approach

- Exact Methods:
  - Computing $\Pr[Y > \gamma]$ exactly is expensive
  - Incorporates pruning techniques.



Our Approach

- Exact Methods:
  - Computing $\Pr[Y > \gamma]$ exactly is expensive
  - Incorporates pruning techniques.
  - Combine the adaptive threshold algorithm for deterministic data when it's applicable.



Our Approach

- Exact Methods:
  - Computing $\Pr[Y > \gamma]$ exactly is expensive
  - Incorporates pruning techniques.
  - Combine the adaptive threshold algorithm for deterministic data when it's applicable.
- Approximate Methods:
  - Replace the exact computation of $\Pr[Y > \gamma]$ using sampleing method (but with the same monitoring instance).

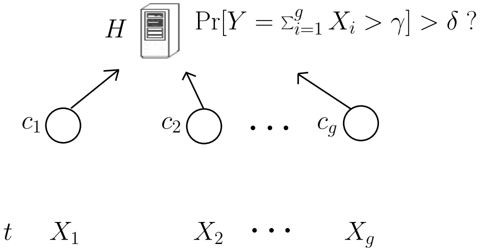
Outline

Baseline method (Madaptive)

- Markov's inequality: $\Pr[Y > \gamma] \leq \frac{\mathbf{E}(Y)}{\gamma}$

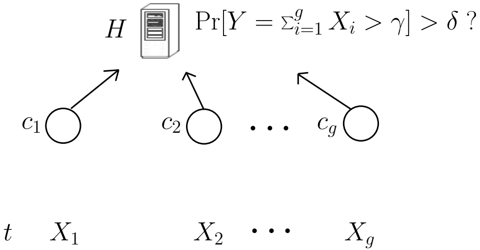
Baseline method (Madaptive)

- Markov's inequality: $\Pr[Y > \gamma] \leq \frac{\mathbf{E}(Y)}{\gamma}$



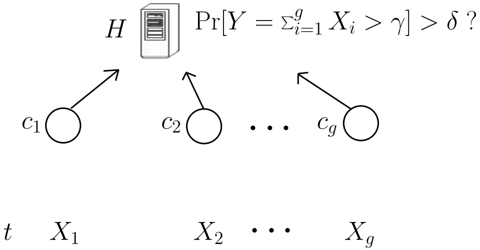
Baseline method (Madaptive)

- Markov's inequality: $\Pr[Y > \gamma] \leq \frac{\mathbf{E}(Y)}{\gamma} < \delta$?



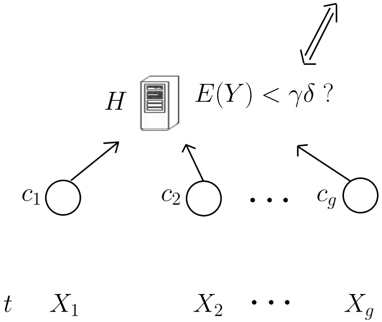
Baseline method (Madaptive)

- Markov's inequality: $\Pr[Y > \gamma] \leq \frac{\mathbf{E}(Y)}{\gamma} < \delta$? $\rightarrow$ *ture* (no alarm)



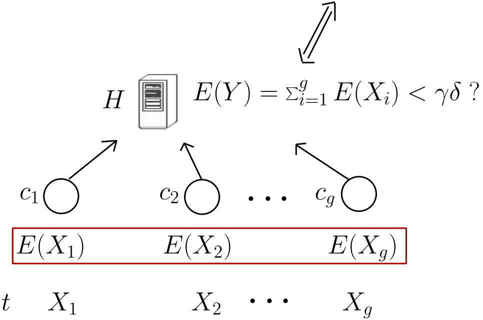
Baseline method (Madaptive)

- Markov's inequality: $\Pr[Y > \gamma] \leq \frac{\mathbf{E}(Y)}{\gamma} < \delta$? $\rightarrow$ *ture* (no alarm)



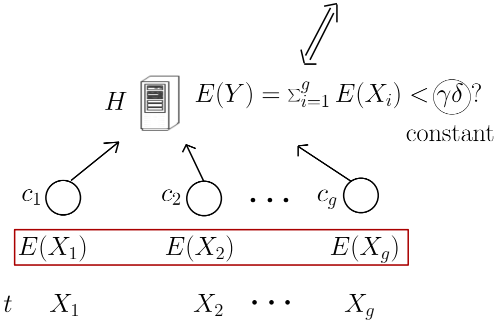
Baseline method (Madaptive)

- Markov's inequality: $\Pr[Y > \gamma] \le \frac{\mathbf{E}(Y)}{\gamma} < \delta$? $\rightarrow$ *ture* (no alarm)



Baseline method (Madaptive)

- Markov's inequality: $\Pr[Y > \gamma] \leq \frac{\mathbf{E}(Y)}{\gamma} < \delta$? $\rightarrow$ *ture* (no alarm)



- Leverage on the adaptive thresholds algorithm for deterministic data

Improved method

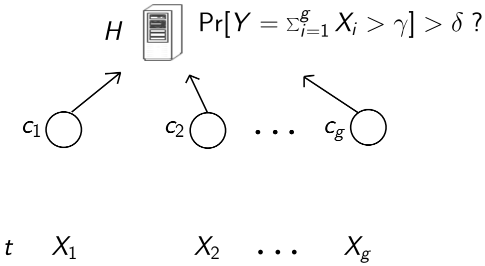
- Combine the Chebyshev bound and Chernoff bound pruning.

Improved method

- Combine the Chebyshev bound and Chernoff bound pruning.
- Chebyshev gives one-sided bound using $\mathbf{E}(X_i)$ and $\mathbf{Var}(X_i)$

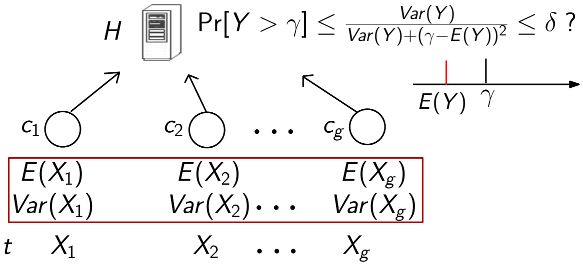
Improved method

- Combine the Chebyshev bound and Chernoff bound pruning.
- Chebyshev gives one-sided bound using $\mathbf{E}(X_i)$ and $\mathbf{Var}(X_i)$



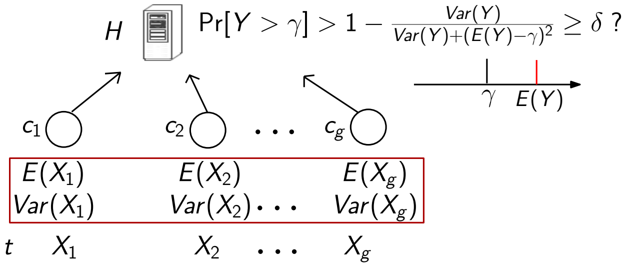
Improved method

- Combine the Chebyshev bound and Chernoff bound pruning.
- Chebyshev gives one-sided bound using $\mathbf{E}(X_i)$ and $\mathbf{Var}(X_i)$



Improved method

- Combine the Chebyshev bound and Chernoff bound pruning.
- Chebyshev gives one-sided bound using $\mathbf{E}(X_i)$ and $\mathbf{Var}(X_i)$

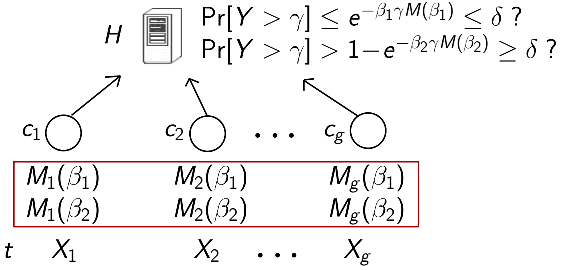


Improved method

- Combine the Chebyshev bound and Chernoff bound pruning.
- Chebyshev gives one-sided bound using $\mathbf{E}(X_i)$ and $\mathbf{Var}(X_i)$
- Chernoff bound using the moment generating function
  - $M(\beta) = \mathbf{E}(e^{\beta Y})$, $M_i(\beta) = \mathbf{E}(e^{\beta X_i})$ for any $\beta \in \mathbb{R}$
  - $M(\beta) = \prod_{i=1}^{g} M_i(\beta)$

Improved method

- Combine the Chebyshev bound and Chernoff bound pruning.
- Chebyshev gives one-sided bound using $\mathbf{E}(X_i)$ and $\mathbf{Var}(X_i)$
- Chernoff bound using the moment generating function
  - $M(\beta) = \mathbf{E}(e^{\beta Y})$, $M_i(\beta) = \mathbf{E}(e^{\beta X_i})$ for any $\beta \in \mathbb{R}$
  - $M(\beta) = \prod_{i=1}^{g} M_i(\beta)$
- $\beta_1 > 0$, Chernoff gives an upper bound
  $\beta_2 < 0$, Chernoff gives a lower bound



Improved Adaptive Method (Iadaptive)

- $\sum_{i=1}^{g} \ln M_i(\beta_1) \leq \ln \delta + \beta_1 \gamma$, (monitoring instance J_1).
- $\sum_{i=1}^{g} \ln M_i(\beta_2) \leq \ln(1-\delta) + \beta_2 \gamma$, (monitoring instance J_2).

Improved Adaptive Method (Iadaptive)

- $\sum_{i=1}^{g} \ln M_i(\beta_1) \leq \ln \delta + \beta_1 \gamma$, (monitoring instance J_1).
- $\sum_{i=1}^{g} \ln M_i(\beta_2) \leq \ln(1-\delta) + \beta_2 \gamma$, (monitoring instance J_2).
- Practical considerations
  - Use the adaptive thresholds algorithm
  - Get a tight upper bound (lower bound)

Improved Adaptive Method (Iadaptive)

- $\sum_{i=1}^{g} \ln M_i(\beta_1) \leq \ln \delta + \beta_1 \gamma$, (monitoring instance J_1).
- $\sum_{i=1}^{g} \ln M_i(\beta_2) \leq \ln(1-\delta) + \beta_2 \gamma$, (monitoring instance J_2).
- Practical considerations
  - Use the adaptive thresholds algorithm
  - Get a tight upper bound (lower bound)
- Approaches
  - Fix the values of β_1 and β_2 in each period of k time instance.
  - Reset the optimal values of β_1 and β_2 periodically.

Improved Adaptive Method (Iadaptive)

- $\sum_{i=1}^{g} \ln M_i(\beta_1) \leq \ln \delta + \beta_1 \gamma$, (monitoring instance J_1).
- $\sum_{i=1}^{g} \ln M_i(\beta_2) \leq \ln(1-\delta) + \beta_2 \gamma$, (monitoring instance J_2).
- Practical considerations
  - Use the adaptive thresholds algorithm
  - Get a tight upper bound (lower bound)
  - Running J_1 and J_2 together is communication expensive.
- Approaches
  - Fix the values of β_1 and β_2 in each period of k time instance.
  - Reset the optimal values of β_1 and β_2 periodically.

Improved Adaptive Method (Iadaptive)

- $\sum_{i=1}^{g} \ln M_i(\beta_1) \leq \ln \delta + \beta_1 \gamma$, (monitoring instance J_1).
- $\sum_{i=1}^{g} \ln M_i(\beta_2) \leq \ln(1-\delta) + \beta_2 \gamma$, (monitoring instance J_2).
- Practical considerations
  - Use the adaptive thresholds algorithm
  - Get a tight upper bound (lower bound)
  - Running J_1 and J_2 together is communication expensive.
- Approaches
  - Fix the values of β_1 and β_2 in each period of k time instance.
  - Reset the optimal values of β_1 and β_2 periodically.
  - Periodically decides which monitoring instance to run

Outline

Our Approach

- Exact Methods:
  - Computing $\Pr[Y > \gamma]$ exactly is expensive in terms of both communication ($O(gT)$) and computation ($O(n^g T)$).
  - Incorporates pruning techniques.
  - Combine the adaptive threshold algorithm for deterministic data when it's applicable.
- Approximate Methods:

Our Approach

- Exact Methods:
  - Computing $\Pr[Y > \gamma]$ exactly is expensive in terms of both communication ($O(gT)$) and computation ($O(n^g T)$).
  - Incorporates pruning techniques.
  - Combine the adaptive threshold algorithm for deterministic data when it's applicable.
- Approximate Methods:
  - We use ϵ-Sampling methods to estimate the condition when monitoring instances fail to make a decision

Our Approach

- Exact Methods:
  - Computing $\Pr[Y > \gamma]$ exactly is expensive in terms of both communication ($O(gT)$) and computation ($O(n^g T)$).
  - Incorporates pruning techniques.
  - Combine the adaptive threshold algorithm for deterministic data when it's applicable.
- Approximate Methods:
  - We use ϵ-Sampling methods to estimate the condition when monitoring instances fail to make a decision
  - Replace the exact computation using sampling based method (but with the same monitoring instance): we get MadapativeS, ImprovedS, IadaptiveS

Random Distributed ε-Sample ($\textsc{RD}\varepsilon\textsc{S}$)

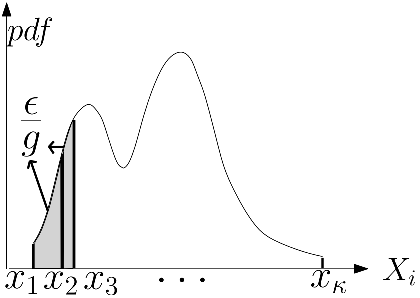
- H asks for a random sample x_i from each client according to the distribution of X_i
- $\Pr[\tilde{Y} = \sum_{i=1}^{g} x_i > \gamma]$ is an unbiased estimate of $\Pr[Y > \gamma]$
- Repeating this sampling $\kappa = O(\frac{1}{\varepsilon^2} \ln \frac{1}{\phi})$ times.
- $\Pr[|\Pr[\tilde{Y} > \gamma] - \Pr[Y > \gamma]| \leq \varepsilon] \geq 1 - \phi$ using $O(\frac{g}{\varepsilon^2} \ln \frac{1}{\phi})$ bytes.

Deterministic Distributed ε-Sample (DDεS)

- Using $\kappa = O(\frac{g}{\varepsilon})$ evenly spaced sample points from each X_i.

Deterministic Distributed ε-Sample (DDεS)

- Using $\kappa = O(\frac{g}{\varepsilon})$ evenly spaced sample points from each X_i.
- $\int_{x=x_j}^{x_{j+1}} \Pr[Xi = x]dx = \frac{\varepsilon}{g}$



Deterministic Distributed ε-Sample (DDεS)

- Using $\kappa = O(\frac{g}{\varepsilon})$ evenly spaced sample points from each X_i.
- $\int_{x=x_j}^{x_{j+1}} \Pr[Xi = x]dx = \frac{\varepsilon}{g}$
- The evaluation space $\Pr[\tilde{Y} > \gamma]$ is in $O(\kappa^g)$

$$
\begin{array}{ll}
c_1 : & S_1\{x_{1,1}, x_{1,2}, \ldots, x_{1,\kappa}\} \\
c_2 : & S_2\{x_{2,1}, x_{2,2}, \ldots, x_{2,\kappa}\} \\
\vdots & \qquad\qquad \vdots \\
c_g : & S_g\{x_{g,1}, x_{g,2}, \ldots, x_{g,\kappa}\}
\end{array}
$$

- Using $\kappa = O(\frac{g}{\varepsilon})$ evenly spaced sample points from each X_i.
- $\int_{x=x_j}^{x_{j+1}} \Pr[Xi = x]dx = \frac{\varepsilon}{g}$
- The evaluation space $\Pr[\tilde{Y} > \gamma]$ is in $O(\kappa^g)$

$$\begin{array}{ll}
c_1 : & S_1\{x_{1,1}, x_{1,2}, \ldots, x_{1,\kappa}\} \\
c_2 : & S_2\{x_{2,1}, x_{2,2}, \ldots, x_{2,\kappa}\} \\
\vdots & \qquad\quad \vdots \\
c_g : & S_g\{x_{g,1}, x_{g,2}, \ldots, x_{g,\kappa}\}
\end{array}$$

Deterministic Distributed ε-Sample (DDεS)

- Using $\kappa = O(\frac{g}{\varepsilon})$ evenly spaced sample points from each X_i.
- $\int_{x=x_j}^{x_{j+1}} \Pr[Xi = x]dx = \frac{\varepsilon}{g}$
- The evaluation space $\Pr[\tilde{Y} > \gamma]$ is in $O(\kappa^g)$
- In practice, $O(\kappa^m)$(e.g., $m = 2$) random selected evaluations.

$$
\begin{array}{ll}
c_1 : & S_1\{x_{1,1}, x_{1,2}, \ldots, x_{1,\kappa}\} \\
c_2 : & S_2\{x_{2,1}, x_{2,2}, \ldots, x_{2,\kappa}\} \\
\vdots & \qquad\quad \vdots \\
c_g : & S_g\{x_{g,1}, x_{g,2}, \ldots, x_{g,\kappa}\}
\end{array}
$$

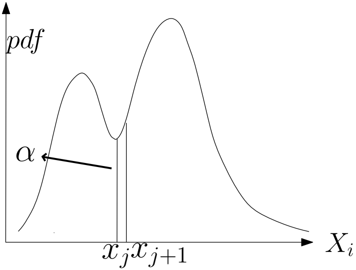
Deterministic Distributed ε-Sample (DDεS)

- Using $\kappa = O(\frac{g}{\varepsilon})$ evenly spaced sample points from each X_i.
- $\int_{x=x_j}^{x_{j+1}} \Pr[Xi = x]dx = \frac{\varepsilon}{g}$
- The evaluation space $\Pr[\tilde{Y} > \gamma]$ is in $O(\kappa^g)$
- In practice, $O(\kappa^m)$(e.g., $m = 2$) random selected evaluations.

$$
\begin{array}{ll}
c_1 : & S_1\{x_{1,1}, x_{1,2}, \ldots, x_{1,\kappa}\} \\
c_2 : & S_2\{x_{2,1}, x_{2,2}, \ldots, x_{2,\kappa}\} \\
\vdots & \qquad\quad \vdots \\
c_g : & S_g\{x_{g,1}, x_{g,2}, \ldots, x_{g,\kappa}\}
\end{array}
$$

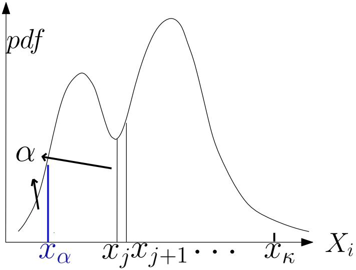
- DDεS gives $|\Pr[\tilde{Y} > \gamma] - \Pr[Y > \gamma]| \leq \varepsilon$ with probability 1 in $O(g^2/\varepsilon)$ bytes.

- $\int_{x=x_{i,j}}^{x_{i,j+1}} Pr[X_i = x]dx = \alpha$

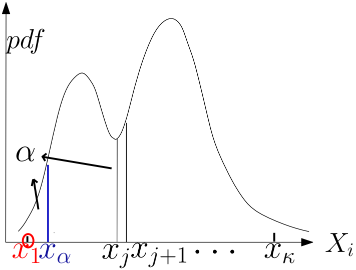


A randomized improvement of DDεS (αDDεS)

- $\int_{x=x_{i,j}}^{x_{i,j+1}} Pr[X_i = x]dx = \alpha$
- Computes x_α where the integral of *pdf* first reaches α

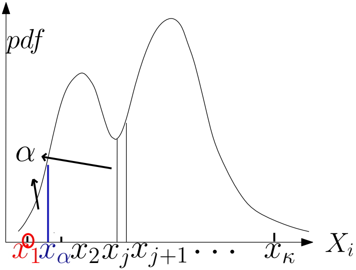


- $\int_{x=x_{i,j}}^{x_{i,j+1}} Pr[X_i = x]dx = \alpha$
- Computes x_α where the integral of *pdf* first reaches α
- Chooses the smallest sample point at random (within x_α).



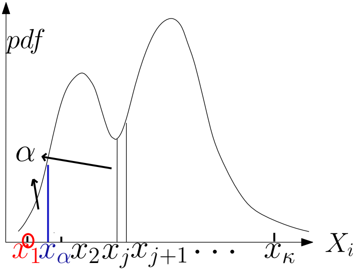
A randomized improvement of DDεS (αDDεS)

- $\int_{x=x_{i,j}}^{x_{i,j+1}} Pr[X_i = x]dx = \alpha$
- Computes x_α where the integral of *pdf* first reaches α
- Chooses the smallest sample point at random (within x_α).



A randomized improvement of DDεS (αDDεS)

- $\int_{x=x_{i,j}}^{x_{i,j+1}} Pr[X_i = x]dx = \alpha$
- Computes x_α where the integral of *pdf* first reaches α
- Chooses the smallest sample point at random (within x_α).



- $\Pr[|\Pr[\tilde{Y} > \gamma] - \Pr[Y > \gamma]| \leq \varepsilon] > 1 - \phi$ in $O(\frac{g}{\varepsilon}\sqrt{2g \ln \frac{2}{\phi}})$ bytes.

Outline

Experiment setup

- A Linux machine with an Intel Xeon CPU at 2.13GHz and 6GB of memory. GMP library are used in calculating $M_i(\beta)$.
- Server-to-client using broadcast and client-to-server using unicast.
- Data sets:
  - Real datasets (11.8 million records in the Wecoma research vessels) from the SAMOS project.
  - Each record contain four measurements: wind direction (WD), wind speed (WS), sound speed (SS), and temperature (TEM), which leads to four single probabilistic attribute datasets.
  - Group the records every τ consecutive seconds and represent it using a pdf.

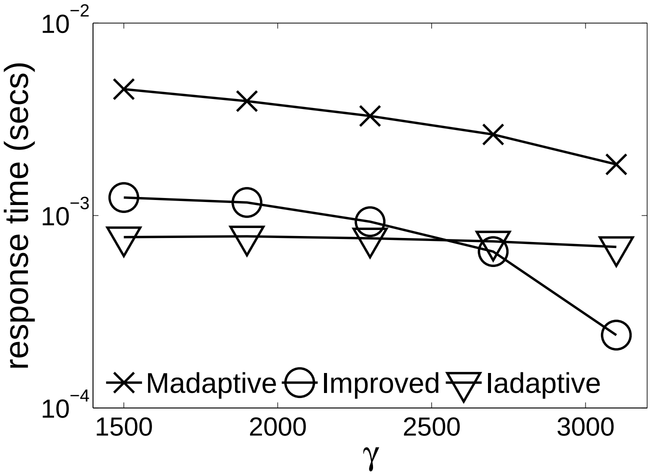
Experiment setup

- The default experimental parameters:

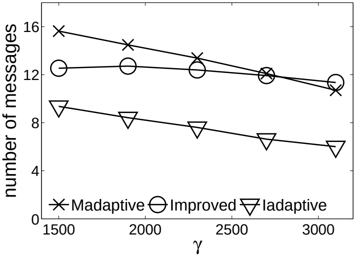
Symbol	Definition	Default Value
τ	grouping interval	300
T	number of time instances	3932
g	number of clients	10
δ	probability threshold	0.7
γ	score threshold	30% alarms (230.g for WD)
κ	sample size per client	30

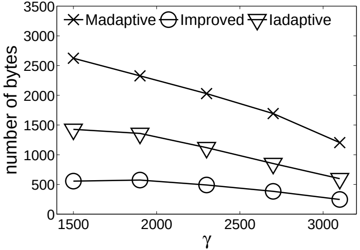
Response time:

- γ: score threshold

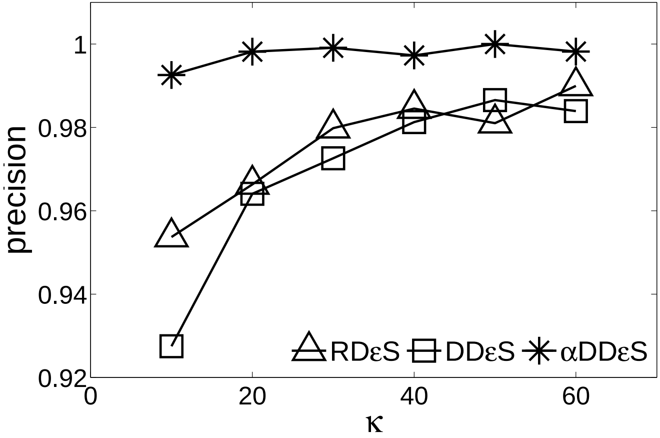


- γ: score threshold



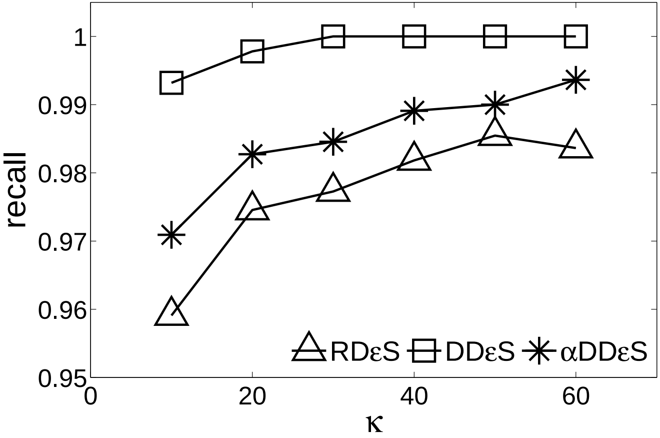


- κ: number of samples

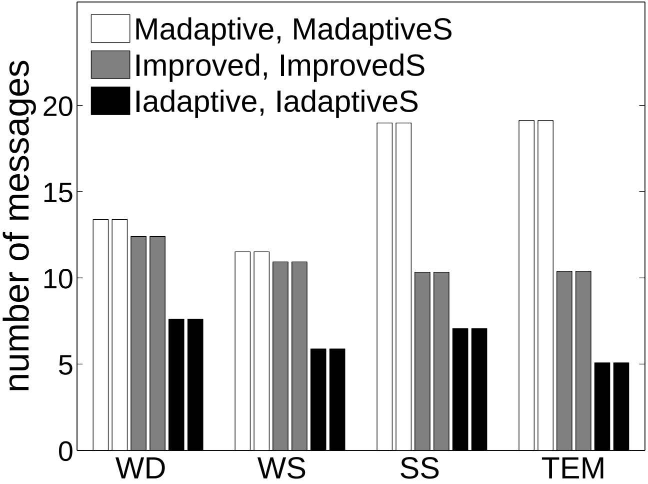


Recall

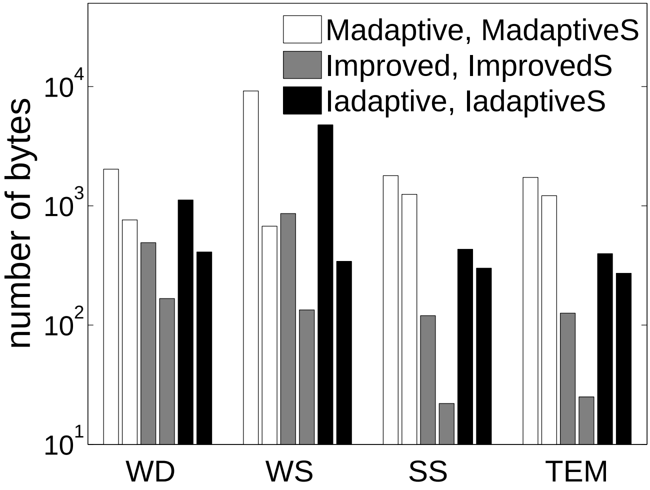
- κ: number of samples



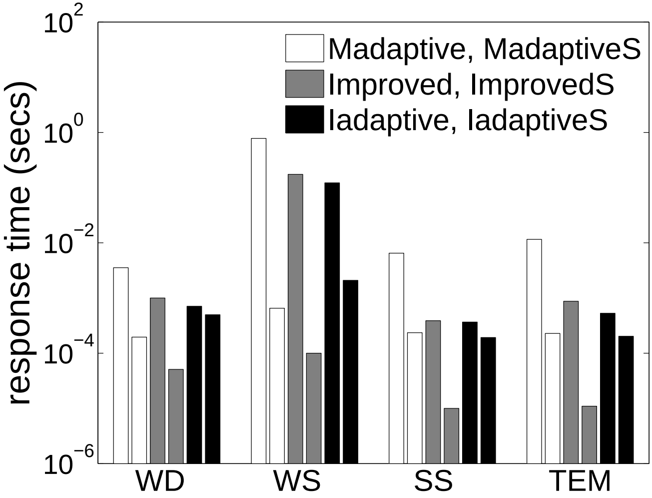
Performance of all methods



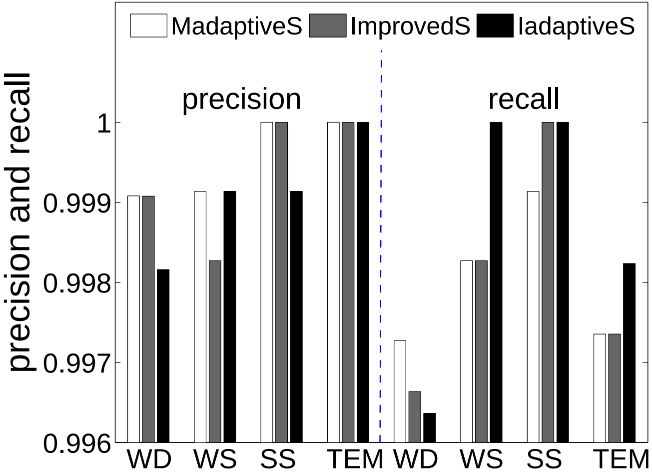
Performance of all methods



Performance of all methods



Performance of all methods



Outline

Conclusion

- Future work:
  - Other aggregation constraints (e.g.,max) beside sum constraint.
  - Extend our study to the hierarchical model that is often used in a sensor network.
  - Handle the case when data from different sites are correlated.

Thank You

Q and A