Real-time Monitoring of Uncertain Data Streams using Probabilistic Similarity*

Honguk Woo, Aloysius K. Mok
Department of Computer Sciences,
The University of Texas at Austin
{honguk, mok}@cs.utexas.edu

Abstract

Data uncertainty is a common problem for the real-time monitoring of data streams. In this paper, we address the issue of efficiently monitoring the satisfaction/violation of user-defined constraints over data streams where the data uncertainty can be probabilistically characterized. We propose a monitoring architecture SPMON *that can incorporate probabilistic models of uncertainty in constraint monitoring. We adapt the concept of data similarity in real-time databases to the processing of uncertain data streams. In doing so, we generalize the data similarity by a new concept* psr *(probabilistic similarity region) that allows us to define similarity relations for probabilistic data with respect to the set of constraints being monitored. This enables the construction of lightweight filters for saving bandwidth. We also show how to efficiently update the filter conditions at run-time.*

1 Introduction

With the adoption of wireless technology in process control applications, the minimization of communication between field devices and host systems is an important issue because of the bandwidth and power limitations in the wireless devices. A well-known strategy is for the resource-limited field devices to report sensor data to the host system only if the data indicates critical state changes of the industrial process under control that the host system must know about [2]. This type of strategy has also been studied to guarantee QoS (quality of service) in the control-theoretic setting [14, 15]. A simple example of this strategy is for a pressure sensor to report its data to the host system if the pressure deviates from some range, say, [500, 1000] psi. This simple rule is straightforward to implement if there is no uncertainty in the sensor data. However, this is no longer the case if the sensor measurements are known to have significant errors due to temporal and spatial uncertainties or other physical limitations, as is the case of many wireless sensor monitoring systems. For example, suppose a pressure sensor obtains a measurement of 990 psi but the measurement can be off by 20 psi. Then the actual pressure can be as low as 970 psi and as high as 1010 psi. In this case it is easy to see that there is a 0.25 probability that the actual pressure has exceeded 1000 psi, assuming that the error probability is uniformly distributed over the [970, 1010] interval of possible pressure values. The decision for the sensor to signal the host system or not can now depend on whether the 0.25 probability is deemed acceptable for the application. For example, given a user-specified threshold δ in probability, the monitoring condition can be specified as: "whether or not the probability that the pressure is in between 500 and 1000 psi is no smaller than δ." Such probability threshold requirement is common for query specification in the uncertain database literature [3, 5, 7].

In general, the processing of sensor data to determine whether or not the host system needs to be signalled can be rather complex if a constraint specifying the critical state change involves more than a single data stream, e.g., a join relation involving pressure data from different locations in a pipe. This problem can be alleviated by an architecture that combines "edge filters" and a "stream processor" such that each data source (or a sensor node) can set up an edge filter to minimize the transmission of data that does not cause some query result to change, and the stream processor can recalibrate the edge filters[1]. This architecture of using edge filters has been studied in various areas such as distributed web caches (e.g., TRAPP system [17]), moving object databases (e.g., safe region based spatial query processing [18, 11]), and stream databases (e.g., adaptive filters [16, 4]). In [16], Olston et. al. investigated the filter adaptation for monitoring continuous aggregation queries over distributed data sources where each source tracks its *exact* numerical value and maintains an *interval* containing the exact value as the filter condition. Employing edge

*This research is partially supported by NSF grant 0613655 and ONR grant N00014-03-1-0705.

[1] In this paper, we use the term *filtering* to denote the suppression of unnecessary data transmission and redundant computation in the context of monitoring continuous queries in stream data management. This differs from the filtering in classical signal processing for noise reduction, which is not our interest in this paper.

1052-8725/07 $25.00 © 2007 IEEE
DOI 10.1109/RTSS.2007.29

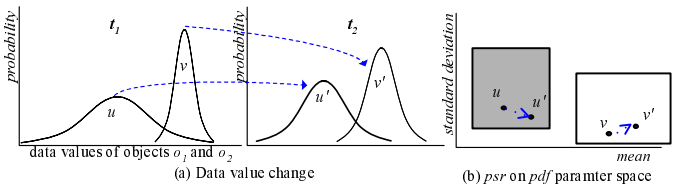


Figure 1. Gaussian *pdf*s and *psr*-based filters

filters is also consistent with emerging industrial standards such as the WirelessHart standard for the process control industry [10].

In this paper, we address the problem of data uncertainty in monitoring constraint-based queries in the stream database context and provide an efficient solution for maintaining edge filters. While aiming at reducing the communication load in a monitoring system like the related work above e.g., [16, 4], our problem is distinct in two aspects that we (1) assume inherent uncertainty in underlying data and (2) consider user-defined constraints with probability thresholds as continuous query specification. Because of these two aspects, the condition of edge filters is formulated based on the constraint specifications and dynamically set by the value changes, and furthermore is made consistent with the uncertainty model. Relating to the uncertainty, we assume that data values at each data source are taken in discrete timesteps and each data value is only given by a *pdf* (probability distribution function).

The efficiency in our method makes explicit use of the concept of *data space similarity* from our earlier work [13, 2]. Two values of a data object are *similar* with respect to a constraint if the satisfaction of the constraint is unaffected by the choice of which value to use in evaluating the constraint. This can be expressed by a bound on the distance between the two values such that two values are similar if their distance does not exceed the bound. In related work, the control-theoretic approach [14, 15] that tracks dynamic objects by adaptive pulling methods also relies on the same bound format for specifying a *temporal coherency* requirement. Usually, such bounds are assumed to be given by the user as a part of the system specification. For our work, we start with a set of user-defined constraints from which we systematically derive *psr* (probabilistic similarity region). Because of the uncertainty in the data values, *psr* is more complex in that it specifies a similarity relation in the parameter space of *pdf* that characterizes uncertain sensor measurements.

Consider the example in Figure 1 (the detail is in Example 1-2 in Section 2). We have a set of user-defined constraints over two objects o_1 and o_2 that have the same type. Assume in this case the data values of o_1 and o_2 are given by the gaussian distribution with different parameters, and accordingly each constraint specification contains a given probability threshold. In figure(a), we have a pair of gaussian distributions u and v that denote the estimate of o_1 and o_2 respectively in timestep t_1. As time progresses to the next timestep t_2, we have new estimates u' and v' for o_1 and o_2. Figure(b) depicts an example of *psr* as rectangles in the 2-dimensional (mean vs. standard deviation) space of gaussian *pdf* parameters. A point in this space denotes a gaussian *pdf*. The gray (white) rectangle, the *psr* of o_1 (o_2), depicts the area where any point (gaussian *pdf*) is similar to u (v) with respect to the given constraints. Notice that *psr* is not given by the user, but is derived from the constraints. The derivation for *psr* will be shown in Section 3 and 4. In figure(b), both estimates of o_1 and o_2 in timestep t_2 are located within their respective *psr*, ensuring that they do not affect the evaluation result of the given constraints, and so they do not have to be forwarded to the stream processor.

While the computation of *psr* for a data object is performed by the stream processor, the filtering of inconsequential new data can be done locally at the data sources by having them maintain their own edge filters (see Figure 2). As pointed out in [18, 16, 8], it is often critical to reduce communication load where the underlying infrastructure is constrained in energy and network resources. In our approach, *psr* realizes the construction of edge filters on probabilistic data, thus enabling suppression of unnecessary data transmission.

1.1 Related Work

In recent years, various adaptive techniques in stream data management have been proposed [12, 16, 18, 4, 1, 19]. Particularly the work in [16, 18, 11, 4] can be categorized as *edge filtering* on remote data sources, as mentioned above. The edge filtering approach exploits the specific query semantics (e.g., numerical aggregation queries with absolute precision requirements [16], spatial range queries [11], and entity-based queries [4]) for efficient filter adaptation. In [11], a *safe region* of objects being tracked is used for efficiently monitoring spatial queries in the moving object database literature. A safe region ensures the result of its associated spatial queries to remain unchanged as long as a newly measured data value does not deviate from the safe region itself. While none of these previous studies in edge filtering considered inherent uncertainty in underlying data, our work specifically addresses the probabilistic models for uncertain data together with the probabilistic semantics for constraint-based queries, and concentrates on the efficient derivation of filter conditions for probabilistic data. It is also important to note that we consider the probabilistic models only for capturing the inherent imprecision in measurement data [3, 7, 9]. Such imprecision is generally incurred by error-prone sensing mechanisms independently from the monitoring process. Thus our work differs from the approximate processing (e.g., [16, 19]) that trades off the data quality for the query processing performance.

The notion of *similarity* has been formalized in the real-time database literature to cope with the dynamics of real-world objects in real-time environments. In [13], Kuo et. al. employed the similarity to provide a clear semantic foundation for relaxing the data consistency requirements on real-time objects and scheduling real-time transactions. In [2], Chen and Mok generalized the similarity relation in the form of predicates on data values to support the flexible query semantics in distributed real-time databases. In comparison, our work adapts the similarity to *pdf* parameter spaces for dealing with uncertain data.

1.2 Contribution

The contributions of this paper are as follows.

- In Section 2, we present our monitoring architecture, called SPMON (**S**imilarity-based **P**robabilistic **MON**itoring), which incorporates data uncertainty in stream data management. In doing so, we specifically formulate the constraint monitoring over uncertain data streams as an edge filtering problem. SPMON supports the efficient suppression of unnecessary data transmission by exploiting the *similarity* of probabilistic data.
- To demonstrate the applicability of SPMON, we illustrate how to efficiently compute *psr* for edge filters at run-time. In particular, we exploit precomputed satisfaction and violation subspaces in the parameter spaces of the uniform and gaussian distributions, considering both static variance in Section 3 and dynamic variance cases in Section 4.

2 SPMON Overview

In this section, we first describe the probabilistic model for uncertain data streams and introduce the probabilistic constraints as our continuous query types. Given the model on data streams and constraints, we then explain how monitoring with uncertain data is formulated as the *psr*-based filtering problem in SPMON.

Figure 2 illustrates the structure of SPMON. The data sources (*ds* in the figure) rely on their edge filters (*EFil*) to determine if new data values need to be forwarded to the stream processor which in turn updates the filtering criteria of the edge filters and returns the evaluation results of monitored constraints. Central to the operation of the edge filter is the concept of *psr*.

2.1 Uncertain Data Stream Model

The source of data is a set of real-valued objects O=$\{o_i\}_{i=1,2,\ldots,|O|}$. In generating data values for the data sources, each object embodies a stochastic process that is indexed by time. Throughout this paper, we make the following assumptions. (1) The value of an object is a function of time. The value can only be estimated at every timestep by a *pdf*. A *pdf* is specified by its parameters as dictated by the distribution type. For example, a gaussian *pdf* is defined by its (mean, standard deviation) pair. We shall use the term *p-data* to denote the set of parameters that characterize a *pdf*, and the term *p-data space* to denote the Cartesian space whose coordinates are the parameters of the *pdf*. (2) Each object generates a sequence of *p-data* as measured by the data sources at each timestep. (3) The data sources are stateless. The *p-data* obtained by a sensor is dependent solely on the true value of the object at the time of measurement and the device characteristics of the sensor.

2.2 Probabilistic Constraints with Thresholds

In this paper, we consider *value*-based constraints[2] of the following two forms:

$$c_R : \dot{r} \leq o_i \leq \bar{r}, \quad c_J : \dot{r} \leq o_i - o_j \leq \bar{r} \qquad (1)$$

where $[\dot{r}, \bar{r}]$ is an interval such that $\dot{r}, \bar{r} \in \mathbb{R}$ and $\dot{r} < \bar{r}$. We refer to c_R and c_J as *range* constraint and *join* constraint respectively. Notice here o_i and o_j refer to the true values of the objects that can only measured statistically by their *p-data*. Given *p-data* u of o_i and v of o_j in timestep t, the satisfaction probability of c_R and c_J are defined respectively as:

$$\begin{aligned} prob(c_R)|_u &= P(\dot{r} \leq X_i \leq \bar{r}), \\ prob(c_J)|_{u,v} &= P(\dot{r} \leq X_i - X_j \leq \bar{r}) \end{aligned}$$

where X_i and X_j denote respectively the *r.v.* (random variables) associated with o_i and o_j in timestep t. Notice that X_i and X_j are represented by their *p-data*.

We now define the probabilistic constraints for continuous query specification as follows. A probabilistic constraint pc is a pair: (c, δ) where c is a constraint and $0 < \delta < 1$ is a confidence threshold in probability. Corresponding to the range and join constraints, the *probabilistic range* and the *probabilistic join* constraint are denoted by the pairs:

$$pc_R : (c_R, \delta) \quad \text{and} \quad pc_J : (c_J, \delta) \qquad (2)$$

A probabilistic range constraint is satisfied (we write: $pc_R|_u$ = *satisfaction*) if its satisfaction probability is not smaller than the given confidence threshold, and it is violated (we write: $pc_R|_u$ = *violation*) otherwise. Likewise, $pc_J|_{u,v}$ = *satisfaction* if $prob(c_J)|_{u,v} \geq \delta$.

Example 1. *Consider a set of probabilistic constraints:* pc_1*:*$(0 \leq o_1 \leq 60, 0.2)$, pc_2*:*$(-20 \leq o_1 \leq 75, 0.6)$, pc_3*:*$(-40 \leq o_1 - o_2 \leq 20, 0.4)$. *And suppose* p-data u *and* v

[2] In [21], we have investigated the *timing* constraint monitoring over uncertain event timestamps using a generalized probabilistic model-based approach. Our work in this paper is complementary to [21] in that we focus on the monitoring of *value*-based constraints on uncertain data attributes of events.

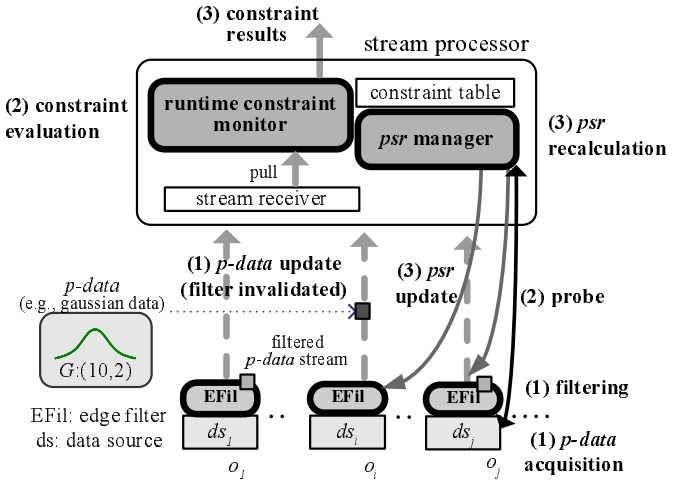


Figure 2. SPMON structure: "*(k) action*" means that the action occurs at the k^{th} phase of the *psr*-based filtering procedure explained in Section 2.3.

are generated for o_1 *and* o_2 *respectively in timestep* t_1*, and given by gaussian distribution, e.g.,* $u \sim N(58, 10^2)$ *and* $v \sim N(79, 2^2)$*. Then we have* $prob(0 \leq o_1 \leq 60)|_u = 0.58$, $prob(-20 \leq o_2 \leq 75)|_v = 0.02$, $prob(-40 \leq o_1 - o_2 \leq 20)|_{u,v} = 0.97$ *(according to the formulae in Section 3.2). Thus in this case* pc_1 *and* pc_3 *are evaluated to be* satisfaction *in timestep* t_1*, but* pc_2 *to be* violation.

2.3 Similarity-based Uncertain Data Filtering

Given the object set O and the probabilistic constraint set C over O, we regard the monitoring process as the updates of the evaluation result (*satisfaction* or *violation*) of each $c_k \in C$ over *p-data* for all $o_i \in O$, which are repeatedly executed in a series of timesteps. In the following, we formulate this monitoring process over uncertain data streams as the edge filtering problem with the goal of the suppression of unnecessary data transmission.

We say that two *p-data* of an object are similar with respect to a probabilistic constraint pc if the evaluation of pc yields the same result with either *p-data*. In the context of data stream processing, a *p-data* that is similar to the one in the previous measurement need not be transmitted to the stream processor. We now define ***psr*** (probabilistic similarity region) as follows:

Definition 1. *Consider objects* o_i*,* o_j *and the probabilistic constraints* pc_R*,* pc_J *in Equation 2. Given* p-data u *of* o_i *and* v *of* o_j*, we define* psr *of* o_i *with respect to* pc_R *by*

$$psr(o_i)|_u^{pc_R} = \mathrm{re_i} \text{ s.t. } u \in \mathrm{re_i} \wedge \forall u' \in \mathrm{re_i}, pc_R|_u = pc_R|_{u'}$$

We define the psr *of* o_i *and* o_j *with respect to* pc_J *by*

$$psr(o_i)|_{u,v}^{pc_J} = \mathrm{re_i}, \ \ psr(o_j)|_{u,v}^{pc_J} = \mathrm{re_j} \quad \text{s.t.}$$
$$u \in \mathrm{re_i} \wedge v \in \mathrm{re_j} \wedge \ \ \forall u' \in \mathrm{re_i} \forall v' \in \mathrm{re_j}, \ pc_J|_{u,v} = pc_J|_{u',v'}$$

where $\mathrm{re_i}$ *and* $\mathrm{re_j}$ *are regions in the* p-data *space of the corresponding objects. We define the overall* psr *of* o_i *by*

$$psr(o_i)|_u = \bigcap_{pc \in C} psr(o_i)|_{u_*}^{pc} \quad (3)$$

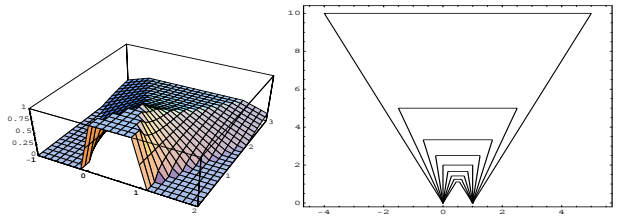
where C *is the constraint set and* u_* *is the* p-data *(i.e., a single* p-data *for range constraints or a pair of* p-data *for join constraints) of objects specified in* pc*. Note that if* pc *does not involve* o_i*, we assume the* $psr(o_i)|_{u_*}^{pc}$ *is the domain of the* p-data *space.*

Suppose we have object o_i, and its corresponding data source ds_i and data stream s_i. As defined above, $psr(o_i)|_u$ denotes the region within which any subsequently obtained *p-data* u' of o_i can be considered unnecessary with respect to monitoring all the constraints (because of Equation 3). Put differently, if u is the last update in s_i, then we can suppress the new updates in s_i by filtering out all the subsequent *p-data* u' of o_i as long as $u' \in psr(o_i)|_u$ holds.

As shown in Figure 2, the monitoring process in SPMON using *psr*-based filters works in the following 3 phases per timestep. Suppose u is the last update in s_i; the filter region of o_i is set as $psr(o_i)|_u$.

- In the first phase, suppose ds_i acquires new *p-data* u' of o_i. If u' does not pass the filter (i.e., $u' \in psr(o_i)|_u$), then u' should be dropped. Otherwise ($u' \notin psr(o_i)|_u$, so the filter is invalidated), u' is put in s_i to be processed by the stream processor. We assume that all data sources complete the *p-data* acquisition, checking *p-data* against the filter condition (filtering), and update to the stream in the first phase.
- In the second phase, a stream processor, on receiving u' of o_i that passes the corresponding filter, retrieves all the associated constraints in the constraint table and re-evaluates the constraints. The constraint table provides the indexing structure of the associated constraints per object. To re-evaluate a probabilistic join constraint that involves the objects o_i and, say o_j (e.g., when the constraint $(10 \leq o_i - o_j \leq 20, 0.8)$ is being monitored), it might be necessary to probe (retrieve) the new *p-data* of o_j if it was filtered out in the previous phase. Once probed, the satisfaction probability of the join constraint can be computed using *p-data* for o_i and o_j to be compared with the threshold.
- In the third phase, the stream processor reports to the host system the results of the constraint evaluation, and finally it re-calculates the filter region of o_i (i.e., $psr(o_i)|_{u'}$) and o_j, and sends the re-calculated filter regions to the respective data sources.

We use a simple predicate form for representing *psr* to minimize the cost of processing continuous data with *psr*-based filters. In our notation, we adopt the following coordinate system for defining rectangular regions (in *pdf* parameter space) on an $\langle x, y \rangle$-plane where x is the horizontal axis



(a) Satisfaction probability (b) Constrained trapezoids

Figure 3. Constrained graph for uniform data: (a) X-axis, Y-axis, and Z-axis represent $-1 \leq m_i \leq 2, 0 \leq w_i \leq 3$, and $prob(0 \leq o_i \leq 1)|_{(m_i,w_i)}$ respectively. (b) X-axis and Y-axis represent $-4.5 \leq m_i \leq 5$ and $0 \leq w_i \leq 10.5$ respectively. Each isosceles trapezoid represents $prob(0 \leq o_i \leq 1)|_{(m_i,w_i)} = \delta$ where $\delta = 0.1, 0.2, \ldots, 0.9$ (from the most outer curve).

and y is the vertical axis. A rectangle having the lower-left corner point $(\dot{x}, \dot{y})$ and the upper-right corner point $(\bar{x}, \bar{y})$ is given by $[\dot{x}, \dot{y}, \bar{x}, \bar{y}]$.

Example 2. *Continuing from Example 1, suppose we have* psr*-based filters on the data sources* o_1 *and* o_2 *for monitoring the constraints* pc_1*,* pc_2*, and* pc_3*. From the given* p-data u *and* v *in timestep* t_1*, we obtain* $psr(o_1)|_u^{pc_1} = [-5.3, 6.4, 65.3, 112.7]$*,* $psr(o_2)|_v^{pc_2} = [74.9, 0.4, \infty, \infty]$*,* $psr(o_1)|_{u,v}^{pc_3} = [48.1, 3.3, 78.9, 36.2]$*,* $psr(o_2)|_{u,v}^{pc_3} = [58, 0, 88.9, 28.2]$*. Then we have* $psr(o_1)|_u = [48.1, 6.4, 65.3, 36.2]$*,* $psr(o_2)|_v = [74.9, 0.4, 88.9, 28.2]$ *by Equation 3. Note that these two rectangles correspond to those in Figure 1(b) where* x*-axis and* y*-axis represent mean and standard deviation respectively. Now consider new* p-data *for* o_1 *and* o_2 *acquired in the next timestep* t_2 *(as illustrated in Figure 1(a)). If such* p-data *are each located in the filter region, that is* $psr(o_1)|_u$ *and* $psr(o_2)|_v$ *above, then they can be dropped from the corresponding streams, e.g., the pair of* $u' \sim N(62, 7^2)$ *and* $v' \sim N(83, 4^2)$ *in* t_2 *can be suppressed. The calculation for this rectangular* psr *will be explained in Section 4.*

In the subsequent sections, we shall show how to calculate *psr* for uncertain data streams that are characterized by uniform or gaussian distribution. The gaussian distribution is common in modeling statistical properties for tracking and monitoring physical objects [6, 9]. The uniform distribution is often used for query processing on uncertain data because of its simplicity for probability computation, analysis and index construction [5].

3 *psr* Formation with Static Variances

3.1 *psr* for Uniform data

We denote *p-data* characterized by the uniform distribution by U_i: (m_i, w_i), noting that $P(X = x) = \frac{1}{w_i}$ if $x \in [m_i - \frac{w_i}{2}, m_i + \frac{w_i}{2}]$ and $P(X = x) = 0$ otherwise, where $m_i, w_i \in \mathbb{R}$, $w_i > 0$, and X is the *r.v.* characterized by U_i. We shall call U_i *uniform data*. Regarding a set of uniform data U_i that satisfy a probabilistic range constraint pc_R: $(\dot{r} \leq o_i \leq \bar{r}, \delta)$ (Equation 2), we obtain the following inequality: $prob(\dot{r} \leq o_i \leq \bar{r})|_{U_i} =$

$$\frac{\max(\min(\bar{r}, m_i + \frac{w_i}{2}) - \max(\dot{r}, m_i - \frac{w_i}{2}), 0)}{w_i} \geq \delta \quad (4)$$

From this inequality, it is possible to derive the satisfaction and violation areas in a 2-dimensional parameter space, so-called $\langle m, w \rangle$-plane[3]. Figure 3(a) shows an example graph of the satisfaction probability with various uniform data and Figure 3(b) depicts its contour graph. Each contour line denotes the satisfaction and violation areas for a specific value of δ. For satisfying pc_R, we have

$$(\delta - \frac{1}{2})w_i + \dot{r} \leq m_i \leq (\frac{1}{2} - \delta)w_i + \bar{r} \wedge w_i \leq \frac{\bar{r} - \dot{r}}{\delta} \quad (5)$$

regarding U_i: (m_i, w_i).

Note that throughout Section 3, the variance of each object o_i is assumed to be time-invariant. For example, in the uniform data model, for all possible uniform data of o_i during the monitoring period, w_i is a constant and thus a single data item for m_i is transmitted, whenever necessary. Accordingly, the filter condition (*psr*) is represented as an *interval* regarding m_i.

3.1.1 Interval *psr* for Range Constraints

Consider pc_R and U_i:(m_i, w_i) of o_i. Corresponding to a specific constant $w_i \leq \frac{\bar{r} - \dot{r}}{\delta}$ and a value of δ, we have the partitioned satisfaction and violation ranges regarding m_i (e.g., from Figure 3(b)). Thus, we have

$$psr(o_i)|_{U_i}^{pc_R} = \begin{cases} (x_u, \infty] & \text{if } m_i > x_u \wedge w_i \leq \frac{\bar{r} - \dot{r}}{\delta} \\ [x_l, x_u] & \text{if } x_l \leq m_i \leq x_u \wedge w_i \leq \frac{\bar{r} - \dot{r}}{\delta} \\ [\text{-}\infty, x_l) & \text{if } m_i < x_l \wedge w_i \leq \frac{\bar{r} - \dot{r}}{\delta} \\ [\text{-}\infty, \infty] & \text{otherwise} \end{cases} \quad (6)$$

where $x_l = (\delta - \frac{1}{2})w_i + \dot{r}$ and $x_u = (\frac{1}{2} - \delta)w_i + \bar{r}$.

3.1.2 *psr* for Join Constraints

We now consider a probabilistic join constraint pc_J (Equation 2) and U_i:(m_i, w_i) of o_i, U_j:(m_j, w_j) of o_j. We have $prob(c_J)|_{U_i,U_j} = \frac{ov}{w_i w_j}$ where ov is the overlapping area by the rectangle $\mathrm{rc_{ij}}$: $[m_i - \frac{w_i}{2}, m_j - \frac{w_j}{2}, m_i + \frac{w_i}{2}, m_j + \frac{w_j}{2}]$ and the constraint c_J: $\dot{r} \leq o_i - o_j \leq \bar{r}$ on the $\langle o_i, o_j \rangle$-plane.

To find the *psr* of o_i and o_j, we take two steps; we first derive the partitioned satisfaction and violation ranges regarding $m_i - m_j$ from pc_J by transforming pc_J to the constraint on $m_i - m_j$, and then get the appropriate range predicate for each m_i and m_j depending on the partitioned ranges and the given uniform data. The detail of this two-step procedure can be found in [20]. For simplicity, suppose we obtain $s_l \leq m_i - m_j \leq s_u$ by the first step. We

[3]The plane of uniform data parameters, middle m and width w.

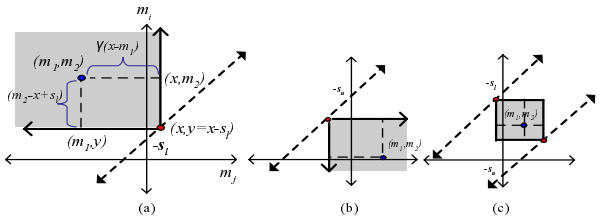


Figure 4. Weighted middle partition: In figure(a), by Equation 7, we get x=$\frac{s_l+\gamma m_i+m_j}{\gamma+1}$ and y=$\frac{\gamma m_i+m_j-\gamma s_l}{\gamma+1}$.

now consider the partitioned area on $\langle m_i, m_j\rangle$-plane. Let m_1 and m_2 be the given specific value of m_i and m_j respectively. First, consider the violation case such that (a) $m_2 > m_1 - s_l$. By constructing a non-closed rectangle that is tangent to the line $m_j = m_i - s_l$ and contains the point (m_1, m_2) on $\langle m_i, m_j\rangle$-plane, we can derive *psr* regarding m_i and m_j. Each side of the rectangle corresponds to *psr*. Although there might exist an infinite number of possible *psr* formations in this case, we can choose a formation that stands a smaller chance of having its filters invalidated by *p-data* in the relatively near future. To do so, we take the tangent point $(x, x - s_l)$ such that

$$\gamma_{m_i,m_j}(x - m_1) = m_2 - x + s_l \tag{7}$$

where $\gamma_{m_i,m_j} > 0$ is a weight parameter[4]. Note that in general γ can be specifically set if the change pattern of objects, e.g., the rate of max (or mean) change per timestep or the rate of filter invalidations, is modeled, and otherwise simply set γ=1. We will discuss our adaptive approach to γ settings in the experiments (e.g., Figure 10). We call this *psr* formation policy, *weighted middle partition* (see Figure 4(a)). In the same sense, we have *psr* formations for the other cases (Figure 4(b)-(c)): (b) $m_2 < m_1 - s_u$ and (c) $s_l \le m_1 - m_2 \le s_u$. Finally, given specific U_i of o_i and U_j of o_j, we have $psr(o_i)|_{U_i}^{pc_J}, psr(o_j)|_{U_j}^{pc_J}$=

$$\begin{cases} [\text{-}\infty, g(s_l)), & (g(-\gamma s_l), \infty] & \text{if } m_j > m_i - s_l \\ (g(s_u), \infty], & [\text{-}\infty, g(-\gamma s_u)) & \text{if } m_j < m_i - s_u \\ [g(s_l), g(s_u)], & [g(-\gamma s_u), g(-\gamma s_l)] & \text{otherwise} \end{cases} \tag{8}$$

where $g(x) = \frac{x+\gamma m_i+m_j}{\gamma+1}$ in case that s_l and s_h exist.

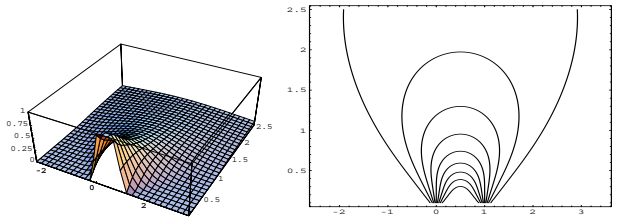
3.2 *psr* for Gaussian data

We denote *p-data* that is characterized by the gaussian distribution by G_i: (μ_i, σ_i), noting that $X \sim N(\mu_i, \sigma_i^2)$ where X is the *r.v.* characterized by G_i. Such G_is will be called *gaussian data*. We represent the standard normal distribution function and its inverse function by

$\Phi(x) = P(z \le x)$ and $\Phi^{-1}(\Phi(x)) = x$ where $z \sim N(0, 1)$.

For simplicity in explanation, we will use, besides pc_R and pc_J, the following constraint with a unit-length range $[0, 1]$, the so-called *unit-length constraint*.

[4] γ_{m_i,m_j} specifies the partitioning weight with respect to the *p-data* parameters related in the satisfaction probability calculation, in this case m_i and m_j. We omit the subscripts and use γ loosely without confusion.



(a) Satisfaction probability (b) Constrained curves

Figure 5. Constrained graph for gaussian data: (a) X-axis represents $-2.5 \le \mu_i \le 3$, Y-axis represents $0 \le \sigma_i \le 2.5$, and Z-axis represents $prob(c_U)|_{(\mu_i,\sigma_i)}$ where c_U is defined in Equation 9. (b) X-axis represents $-2.5 \le \mu_i \le 3.5$, Y-axis represents $0 \le \sigma_i \le 2.5$, and the curves represent $prob(c_U)|_{(\mu_i,\sigma_i)} = \delta$ where $\delta = 0.1, 0.2, \ldots, 0.9$ (from the most outer curve).

$$c_U : 0 \le o_i \le 1, \quad pc_U : (c_U, \delta) \tag{9}$$

Given a set of gaussian data satisfying pc_U, we obtain a non-linear curve on the 2-dimensional mean and standard deviation parameter space, i.e., $\langle \mu, \sigma\rangle$-plane, by using the following inequality.

$$prob(c_U)|_{G_i} = \Phi(\frac{1-\mu_i}{\sigma_i}) - \Phi(\frac{0-\mu_i}{\sigma_i}) \ge \delta \tag{10}$$

Although it might appear simple to locate individual points on the constraint curve, there is no analytical solution, i.e., we do not have a closed form solution to obtain an x that satifies $prob(c_U)|_{(\mu_i,x)} = \delta$ for specific values of μ_i and δ except for μ_i=$\frac{1}{2}$. Figure 5(a) shows an example graph of the satisfaction probability with various gaussian data and Figure 5(b) depicts its contour lines, which can be numerically found. Each contour line corresponds to the constrained curve by Equation 10 and thus describes the satisfaction and violation areas for pc_U with a specific value of δ, similarly as shown in Figure 3(b).

Lemma 1. *Given a range constraint c_R: $\dot{r} \le o_i \le \bar{r}$ and gaussian data (μ_i, σ_i) of o_i, we have*

$$prob(c_R)|_{(\mu_i,\sigma_i)} = prob(0 \le o_i \le 1)|_{(\mu_u,\sigma_u)}$$

where $\mu_u = \frac{\mu_i - \dot{r}}{\bar{r} - \dot{r}}$ and $\sigma_u = \frac{\sigma_i}{\bar{r} - \dot{r}}$.

Lemma 1 allows us to transform a constrained curve of the unit-length constraint to that of any arbitrary range constraint with the same value of δ. Furthermore, we can exploit the symmetry of constrained curves. Note that all the curves in Figure 5(b) are symmetric about $\frac{\dot{r}+\bar{r}}{2}$ = $\frac{1}{2}$, to be consistent with the following lemma.

Lemma 2. *Given a range constraint c_R: $\dot{r} \le o_i \le \bar{r}$ and gaussian data (μ_i, σ_i) of o_i, we have*

$$prob(c_R)|_{(\mu_i,\sigma_i)} = prob(c_R)|_{(\dot{r}+\bar{r}-\mu_i,\sigma_i)}.$$

3.2.1 Interval *psr* for Range Constraints

Similarly as in the case of uniform data, *psr* for gaussian data with static variances is given by an interval regarding μ. However, such an interval must be found numerically, unlike uniform data. For efficiently updating *psr*-based filters at run-time, we can pre-calculate a table, named $\mathcal{G}$-table, that contains the approximate points of the constrained curve (e.g., derived by Equation 10) for various values of δ of pc_U. This requires using a numerical method. In particular, to prune unnecessary sampling points in our numerical calculation, we use a rectangle tightly enclosing the constrained curve. Such rectangle is called a *minimum violation rectangle*. The implementation of our numerical method can be found in [20]. In the following, suppose we have the minimum violation rectangle for pc_U, denoted by

$$\mathrm{rc_v} : [\dot{\mu}_v, \dot{\sigma}_v, \bar{\mu}_v, \bar{\sigma}_v] \tag{11}$$

and the following auxiliary function that provides the interface to the $\mathcal{G}$-table.

Definition 2. *For the unit-length constraint pc_U:(c_U, δ), we define the constrained curve function:*

$$\mathcal{G}(x)|_\delta = \mu_i \text{ s.t. } prob(c_U)|_{(\mu_i,x)} = \delta \wedge \mu_i \geq \frac{1}{2} \tag{12}$$

for $x \geq 0$. Note that given $\mu_i = \mathcal{G}(\sigma_i)|_\delta$, we have $prob(c_U)|_{(1-\mu_i,\sigma_i)} = \delta$ by Lemma 2.

It is obvious that given a specific value of $\sigma_i \leq \bar{\sigma}_v$, we have the partitioned satisfaction and violation ranges regarding μ_i. These ranges immediately specify *psr*, and thus

$$psr(o_i)|_{G_i}^{pc_u} = \begin{cases} (x, \infty] & \text{if } \mu_i > x \wedge \sigma_i \leq \bar{\sigma}_v \\ [\text{-}\infty, 1-x) & \text{if } \mu_i < 1-x \wedge \sigma_i \leq \bar{\sigma}_v \\ [1-x, x] & \text{if } 1-x \leq \mu_i \leq x \wedge \sigma_i \leq \bar{\sigma}_v \\ [\text{-}\infty, \infty] & \text{otherwise} \end{cases} \tag{13}$$

where $x = \mathcal{G}(\sigma_i)|_\delta$.

Generalization for arbitrary ranges. We have shown *psr* formation for the unit-length constraint pc_U above. For an arbitrary range $[\bar{r}, \dot{r}]$, the *psr* with respect to pc_U can be adjusted by Lemma 1. For example, from (i) calculated by Equation 13, we obtain (ii) below where $l = \bar{r} - \dot{r}$.

$$\text{(i) } psr(o_i)|_{(\frac{\mu_i - \dot{r}}{l}, \frac{\sigma_i}{l})}^{pc_U} = [\dot{\mu}, \bar{\mu}], \text{ (ii) } psr(o_i)_{(\mu_i,\sigma_i)}^{pc_R} = [l\dot{\mu}+\dot{r}, l\bar{\mu}+\dot{r}] \tag{14}$$

3.2.2 *psr* for Join Constraints

Based on the linear property of the difference of independent gaussian *r.v.*, that is, given G_i:(μ_i, σ_i) and G_j:(μ_j, σ_j),

$$prob(c_J)|_{G_i,G_j} = \Phi(\frac{\bar{r} - \mu_i + \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}) - \Phi(\frac{\dot{r} - \mu_i + \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}),$$

it is possible to find the partitioned ranges regarding $\mu_i - \mu_j$ by using gaussian data G_k: $(\mu_i - \mu_j, \sqrt{\sigma_i^2 + \sigma_j^2})$ and Equation 13-14. This allows us to find the satisfaction range of $\mu_i - \mu_j$, denoted as $s_l \leq \mu_i - \mu_j \leq s_u$. Assume a weight

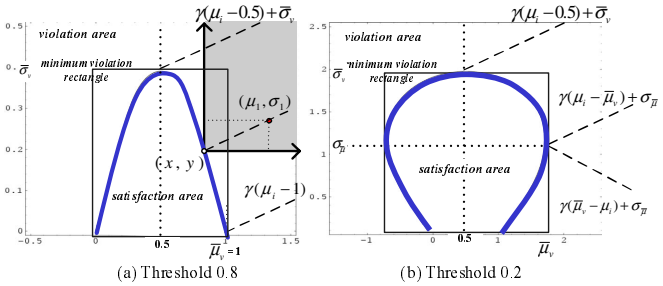


Figure 6. *psr* for pc_U: X-axis and Y-axis represent μ_i and σ_i. The solid line rectangle denotes the minimum violation rectangle. The gray rectangle in figure(a) depicts the third case *psr* in Equation 15 for the given specific gaussian data (μ_1, σ_1).

parameter γ and let $g(x) = \frac{x+\gamma\mu_i+\mu_j}{\gamma+1}$. By the policy of weighted middle partition, then, the respective *psr* of o_i and o_j are given same as Equation 8 except for here using μ_i, μ_j instead of m_i, m_j.

4 *psr* Formation with Dynamic Variances

Thus far, we have focused on *psr* formation with static variances. The static variance assumption is normally suitable for modeling the tracking of object values that are not highly dynamic i.e., they do not change drastically in a short time. In reality, however, there are situations where the uncertainty degree of an object fluctuates for subsequent timesteps and so it should be modeled as a variable. In this section, we consider the gaussian data streams and focus on the *rectangular psr* formation, incorporating the continuous updates of both μ and σ parameters to the *psr* formation and filter construction. In addition to the minimum violation rectangle $\mathrm{rc_v}$ (Equation 11), we notate $\sigma_{\bar{\mu}}$ such that the point $(\bar{\mu}_v, \sigma_{\bar{\mu}})$ is located on $\mathrm{rc_v}$ (see Figure 6(b); $(\bar{\mu}_v, \sigma_{\bar{\mu}})$ is called max-μ point). And let us denote *psr* (e.g., $psr(o_i)|_{G_i}^{pc_U}$) by a rectangle $\mathrm{rc_{psr}}$ on $\langle\mu, \sigma\rangle$-plane. We will then show how to determine $\mathrm{rc_{psr}}$. Because of the symmetric property of Lemma 2, we only consider the case $\mu_i \geq \frac{1}{2}$.

4.1 Rectangular *psr* Formation

Here suppose $pc_U|_{G_i}$= *violation*. First, consider the case of $0.5 \leq \delta < 1$. As shown in Figure 6(a), the constrained curve (i.e., $\mathcal{G}(.)$) is monotonically decreasing with respect to σ_i. Thus, in this case, we prefer $\mathrm{rc_{psr}}$ the lower-left corner (x, y) of which is tangent to the constrained curve, so $\mathrm{rc_{psr}}$ is in the form $[x, y, \infty, \infty]$. As describe before, such (x, y) is determined based on the weighted middle partition; if (x, y) cannot exist by the relation of $\mathrm{rc_v}$ and the given gaussian data, we take either the upper-side or the right-side of $\mathrm{rc_v}$ instead. Then, given specific G_i such that $pc_U|_{G_i}$= *violation*, we have (x, y) =

$$\begin{cases} (\text{-}\infty, \bar{\sigma}_v) & \text{if } \gamma(\mu_i - \frac{1}{2}) + \bar{\sigma}_v \leq \sigma_i \\ (\bar{\mu}_v, 0) & \text{if } \gamma(\mu_i - \bar{\mu}_v) \geq \sigma_i \\ (\mathcal{G}(y)|_\delta, y = \gamma(\mathcal{G}(y)|_\delta - \mu_i) + \sigma_i) & \text{otherwise} \end{cases} \tag{15}$$

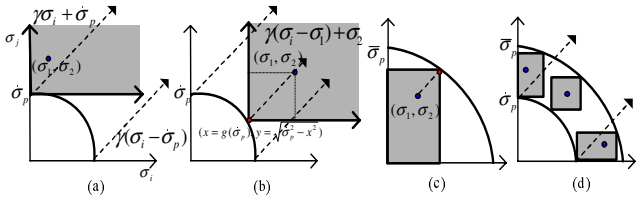


Figure 7. *psr* for σ part: Each gray rectangle with solid lines denotes the range predicates (or *psr* part) on σ_i and σ_j depending on the given specific σ pair (σ_1, σ_2).

where γ is a weight parameter for modeling the change pattern of o_i (e.g., the change ratio between gaussian parameters).

Second, consider the other case $0 < \delta < 0.5$. Different from the previous case, $\mathcal{G}(.)$ is convex around the max-μ point $(\bar{\mu}_v, \sigma_{\bar{\mu}})$ as shown in Figure 6(b). Due to the space limitation, we omit the further detail of this case and the satisfaction case (see [20] for these cases).

Generalization for arbitrary ranges. Equation 14 is extended for rectangular *psr*, $psr(o_i)^{pc_R}_{(\mu_i,\sigma_i)}$ as in (ii) below.

$$\text{(i) } psr(o_i)|^{pc_U}_{(\frac{\mu_i-\dot{r}}{l}, \frac{\sigma_i}{l})} = [\dot{\mu}, \dot{\sigma}, \bar{\mu}, \bar{\sigma}], \text{ (ii) } [l\dot{\mu} + \dot{r}, l\dot{\sigma}, l\bar{\mu} + \dot{r}, l\bar{\sigma}] \tag{16}$$

4.2 *psr* for Join Constraints

By the property of gaussian data explained in Section 3.2.2, given G_i of o_i and G_j of o_j, it is possible to calculate the intermediate *psr* form regarding $\mu_i - \mu_j$ and $\sqrt{\sigma_i^2 + \sigma_j^2}$, from which the respective *psr* of o_i and o_j can be derived. Suppose we obtain such intermediate *psr* as a rectangle $[\dot{\mu}_p, \dot{\sigma}_p, \bar{\mu}_p, \bar{\sigma}_p]$ by the formulae in Section 4.1. Since we have presented our method to derive the respective *psr* part for μ_i and μ_j in Section 3.2.2, here we explain its complementary part. From the condition $\dot{\sigma}_p \le \sqrt{\sigma_i^2 + \sigma_j^2} \le \bar{\sigma}_p$ given by the aforementioned intermediate *psr*, we derive the individual range predicates on each σ_i and σ_j. Note that combining the range predicate on σ_i and that on μ_i completes the *psr* of o_i. In the following, assume a weight parameter γ for modeling the difference of the change patterns of σ_i and σ_j, supposing given specific σ_1 of o_i and σ_2 of o_j. For notational simplicity, let

$$g(x) = \frac{\gamma^2\sigma_1 - \gamma\sigma_2 + \sqrt{x^2 - (\sigma_2 + (x-\sigma_1)\gamma)(\sigma_2 - (\sigma_1 + x)\gamma)}}{1+\gamma^2}.$$

First, if $\dot{\sigma}_p > 0$ and $\bar{\sigma}_p = \infty$, we have

$$\sigma_i \ge x \text{ and } \sigma_j \ge \sqrt{{\dot{\sigma_p}}^2 - x^2}$$

where

$$x = \begin{cases} 0 & \text{if } \gamma\sigma_1 + \dot{\sigma}_p \le \sigma_2 \\ g(\dot{\sigma}_p) & \text{if } \gamma(\sigma_1 - \dot{\sigma}_p) < \sigma_2 < \gamma\sigma_1 + \dot{\sigma}_p \\ \dot{\sigma}_p & \text{otherwise.} \end{cases}$$

Note that Figure 7(a)-(b) show the different *psr* formations according to the first two cases in the above equation. Due to the space limitation, we omit the detail including the other cases corresponding to the examples in figure(c)-(d).

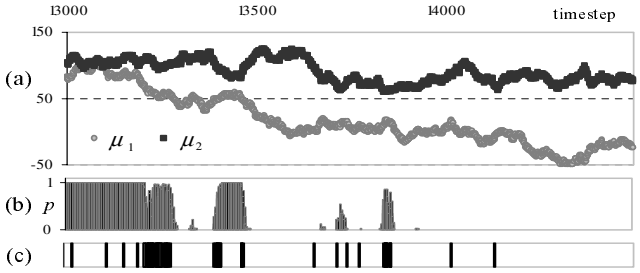


Figure 8. Stream monitoring example: X-axis represents timesteps from 13K to 14.5K. Y-axis represents (a) the changes of μ_1 (for o_1 with static σ_1=4) and μ_2 by the random walk model where λ=3 (Equation 17), (b) satisfaction probability of c_1, and (c) the stream suppression by *psr*-based filters where each vertical line corresponds to a message.

5 Evaluation

To evaluate the effectiveness of our approach, we implemented the SPMON simulator and performed several experiments using the parameters in Table 1.

Parameter	Description
nt	number of timesteps (= 50K)
no	number of tracked objects
co	number of constraints per object
λ	max change (of a *p-data* parameter) per timestep

Table 1. Experiment parameters

For uncertain data streams, we used a random walk model to produce continuous streams of *p-data*, uniform or gaussian data, where the possible value difference of subsequent *p-data* parameters can be specified. In all cases, we used stream data sets of 50K timesteps.

The experiments in Figure 8-10 take a join constraint and gaussian data streams with static variances. For example, Figure 8(a)-(b) show part of gaussian data streams of o_1, o_2 where their variance is constant, that is σ_1=σ_2=4, and the corresponding satisfaction probability of a join constraint c_1:$(-50 \le o_1 - o_2 \le 50)$. Each stream was generated by a random walk model, i.e.,

$$\mu_i^{t+1} = \mu_i^t + \text{U}(-\lambda, \lambda) \tag{17}$$

where μ_i^t denotes μ_i in timestep t, $\lambda \in \mathbb{R}^+$ denotes the maximum of a random walk, and $\text{U}(-\lambda, \lambda)$ denotes uniform distribution on the interval $[-\lambda, \lambda]$. Given the threshold δ=0.8, figure(c) demonstrates the benefit of SPMON (by Section 3.2.2 where we set γ=1) such that update messages during most timesteps were suppressed while monitoring the probabilistic constraint pc_1:$(c_1, 0.8)$.

As a performance metric, we compare the number of messages by SPMON to that by a traditional *central monitor* where *p-data* are transmitted to a stream processor in every timestep: msg (= message transmission rate) = $\frac{nm}{\text{nt}\times\text{no}}$ where nm is the number of messages. For messages in SPMON,

$$nm = nivd + 2nprb + nudt \tag{18}$$

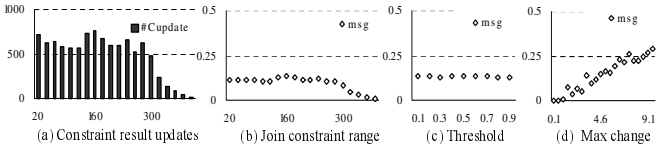


Figure 9. msg with various settings: Figure(a) shows the number of constraint result updates depending on the join constraint range size. In figures(b)-(d), Y-axis represents msg and X-axis represents (b) join constraint range size rs, (c) threshold δ, and (d) max change per timestep λ. Figures(b)-(d) show msg of monitoring $(\frac{-rs}{2} \leq o_1 - o_2 \leq \frac{rs}{2}, \delta)$ where (b) δ=0.8, λ=3, (c) rs=100, λ=3, (d) rs=100, δ=0.8. We set γ=1 for all the cases.

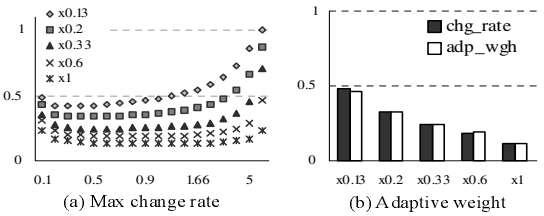


Figure 10. Adaptive weight: Y-axis represents msg of monitoring $(-50 \leq o_1 - o_2 \leq 50, 0.8)$ and X-axis represents (a) γ, (b) max change rate $\frac{\lambda_2}{\lambda_1}$.

where $nivd$ denotes the number of *psr*-based filters that are invalidated by *p-data*; $nprb$ denotes the number of probes initiated by a stream processor to retrieve the *p-data* of the other object in the join when the filter of one object is invalidated; and $nudt$ denotes the number of *psr*-based filter updates. Recall that the re-evaluation of a join constraint upon invalidation by the *p-data* of an object o_i passing through its filter may require probing to the other object in the join in which o_i appears. Owing to these probes (more specifically, probing request and reply messages) and filter update messages, in general, query-aware filtering methods are not always of benefit to message reduction. We will show the effect of the overhead in Figure 11, in some cases with relatively large λ and co. It is also important to note that one cannot simply use a variant of *central monitor* that uses "constant" filter ranges for our problem, since even a minimum value change of a *p-data* parameter in the domain of real numbers may require the revalidation of its associated constraints; in fact, this is also a reason why *monitoring* specific continuous queries (e.g., probabilistic constraints in this paper, spatial queries [18]) is different from *tracking* time-varying data [14, 15] which is independent of query types, although both often aim at the same goal of low communication overhead.

Single join constraint. Figure 9 shows msg of monitoring a single join constraint. As expected and shown in figures(a)-(b), msg is heavily dependent on the degree of fluctuation in the results of constraint evaluation over time; more dynamicity in the tracked objects (i.e., larger λ) is likely to change the constraint results rapidly and accordingly break *psr*-based filters more, thus resulting in larger msg as shown in the figure(d).

As explained in the previous sections, *psr* consists of simple predicates on *p-data* parameters and our *psr* formation formulae contain weight parameters γ. Figure 10 demonstrates the impact of this γ setting against streams with different properties. For the different stream properties, we created streams by increasing the max change of μ_1 per timestep (denoted by λ_1) and setting λ_2=3 constantly, e.g., in figure(a), x0.13 denotes the stream being generated with $\frac{\lambda_2}{\lambda_1}$ = 0.13. In figure(a), we observe that setting γ close to $\frac{\lambda_2}{\lambda_1}$ yields low msg. The reason is that such γ enables the *weighted* range partitions in *psr* formation to be consistent with the *skewness* of random walks for each *p-data* parameter. However, such λs for real uncertain streams may not be *a priori* known. Thus, we tested a simple heuristics named *adaptive weight*: for each *p-data* parameter, the number of time its predicate as part of the *psr*-based filter is broken in a sliding window (e.g., during the 20 recent timesteps) is used instead of λs. In figure(b), we observe the effectiveness of our adaptive method, that is, its msg (denoted by adp_wgt) is fairly competitive in comparison with the best case when $\frac{\lambda_2}{\lambda_1}$ is explicitly used (denoted by chg_rate). Moreover, adaptive methods normally can remain promising under the condition of time-varying stream properties.

Multiple constraints. Thus far, we have studied the performance issue with a single constraint using *psr*. Two major observations were that (1) msg is mainly influenced by the dynamicity of *p-data* and (2) our adaptive weight policy works quite well, not needing *a priori* knowledge of stream properties. Here we discuss the case of *multiple* constraints with dynamic variances (for gaussian data streams in Section 4). We created different sets of objects and constraints randomly using given values for the parameters no, co, the range size of each constraint $rs \in [50, 120]$, and the threshold $\delta \in (0, 1)$. Given co=k, we created $\lceil \frac{k}{2} \rceil$ range constraints and $\lfloor \frac{k}{2} \rfloor$ join constraints for each object. We also created different sets of data streams using given λ of μ and σ for each object. Figure 11 shows that msg is also influenced by co while it is not directly influenced by the size of object or constraint set as long as co is same. This is because *psr* of an object is achieved by the intersection of *psr* with respect to all the constraints being imposed on the object. In other word, a filter region is likely to shrink more along with larger co, implying that the benefit of SPMON could increase with small co and thus SPMON must be used conservatively otherwise; e.g., it seems advantageous to use SPMON for the cases where co$\leq$ 4 and $\lambda < 3$. Finally we note that adaptiveness of SPMON such as combining the *psr*-based filter and non-filtering of *central monitor* for different stream and constraint properties is an interesting is-

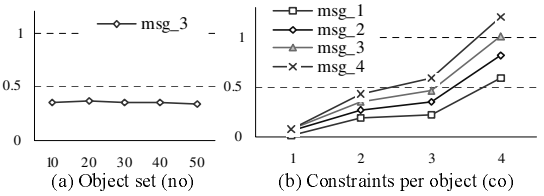


Figure 11. Multiple constraints: Y-axis represents msg. msg_n denotes the different streams of which λ of μ and σ are given n and $\frac{n}{2}$ respectively. X-axis represents (a) no and (b) co. We set (a) co=2 and (b) no=10.

sue but is not included here due to the space limit.

6 Conclusion

As usage of sensor-based real-time monitoring systems grows, probabilistic data processing based on statistical modeling techniques has gained much attention in both the stream database and real-time database literature [5, 7, 12]. This paper introduces SPMON, a monitoring architecture that incorporates data value uncertainty in monitoring constraint-based queries. In particular, we generalize the notion of data similarity to cover data objects whose values can only be determined probabilistically. To the best of our knowledge, SPMON is the first to attempt monitoring and filtering of uncertain data streams in *pdf* parameter spaces. As shown in the previous sections, the filter conditions based on the new *psr* concept can be systematically derived for well-known distributions. Our future work includes investigating a unified framework for monitoring uncertain events that is able to cover two different types of uncertainty: event timing in [21] and data value in this work. We also plan to explore complex constraint types in our monitoring queries, e.g., constraints on arbitrary aggregate (linear or non-linear) functions of multiple uncertain data streams.

Acknowledgement. The authors would like to thank prof. Steve Liu for his suggestions on the revision of this paper.

References

[1] M. Amirijoo, N. Chaufette, J. Hansson, S. H. Son, and S. Gunnarsson. Generalized Performance Management of Multi-Class Real-Time Imprecise Data Services. In *Proc. of IEEE Real-Time Systems Symposium(RTSS)*, pages 38–49, December 2005.

[2] D. Chen and A. K. Mok. SRDE: Application of Data Similarity to Process Control. In *Proc. of IEEE Real-Time Systems Symposium(RTSS)*, pages 136–145, December 1999.

[3] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating Probabilistic Queries over Imprecise Data. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 551–562, 2003.

[4] R. Cheng, B. Kao, S. Prabhakar, A. Kwan, and Y. Tu. Adaptive Stream Filters for Entity-based Queries with Non-value Tolerance. In *Proc. of the International Conference on Very Large Data Bases(VLDB)*, pages 37–48, August 2005.

[5] R. Cheng, Y. Xia, , S. Prabhakar, R. Shah, and J. S.Vitter. Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data. In *Proc. of the International Conference on Very Large Data Bases(VLDB)*, pages 876–887, August 2004.

[6] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model Driven Data Acquisition in Sensor Networks. In *Proc. of the International Conference on Very Large Data Bases(VLDB)*, pages 588–599, 2004.

[7] A. Deshpande and S. Madden. MauveDB: Supporting Model-based User Views in Database Systems. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 73–84, June 2006.

[8] Q. Han and N. Venkatasubramanian. Addressing Timeliness/Accuracy/Cost Tradeoffs in Information Collection for Dynamic Environments. In *Proc. of IEEE Real-Time Systems Symposium(RTSS)*, pages 108–117, December 2003.

[9] S. Han, E. Chan, R. Cheng, and K.-Y. Lam. A statistics-based sensor selection scheme for continuous probabilistic queries in sensor networks. *Real-Time Syst.*, 35(1):33–58, 2007.

[10] HART Communication Foundation. WirelessHART Specification Released for Approval. www.hartcomm2.org.

[11] H. Hu, J. Xu, and D. L. Lee. A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 479–490, June 2005.

[12] A. Jain, E. Y. Chang, and Y. F. Wang. Adaptive Stream Resource Management using Kalman Filters. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 11–22, June 2004.

[13] T. W. Kuo and A. K. Mok. Real-Time Data Semantics and Similarity-Based Concurrency Control. *IEEE Transactions on Computers*, 49(11):1241–1254, 2000.

[14] R. Majumdar, K. M. Moudgalya, and K. Ramamritham. Adaptive Coherency Maintenance Techniques for Time-Varying Data. In *Proc. of IEEE Real-Time Systems Symposium(RTSS)*, pages 98–107, December 2003.

[15] R. Majumdar, K. Ramamritham, R. Banavar, and K. Moudgalya. Disseminating Dynamic Data with QoS Guarantee in a Wide Area Network: A Practical Control Theoretic Approach. In *Proc. of Real-Time Technology and Applications Symposium(RTAS)*, pages 510–517, May 2004.

[16] C. Olston, J. Jiang, and J. Widom. Adaptive Filters for Continuous Queries over Distributed Data Streams. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 563–574, June 2003.

[17] C. Olston and J. Widom. Offering a Precision-Performance Tradeoff for Aggregation Queries over Replicated Data. In *Proc. of the International Conference on Very Large Data Bases(VLDB)*, pages 144–155, September 2000.

[18] S. Prabhakar, X. Xia, D. V. Kalashnikov, W. G. Aref, and S. E. Hambrusch. Query Indexing and Velocity Constrained Indexing: Scalable Techniques for Continuous Queries on Moving Objects. *IEEE Transactions on Computers*, 51(10):1124–1140, 2002.

[19] Y. Wei, V. Prasad, S. H. Son, and J. A. Stankovic. Prediction-Based QoS Management for Real-Time Data Streams. In *Proc. of IEEE Real-Time Systems Symposium(RTSS)*, pages 344–355, December 2006.

[20] H. Woo and A. K. Mok. Real-time Monitoring of Uncertain Data Streams using Probabilistic Similarity. Technical report, University of Texas at Austin, www.cs.utexas.edu/users/honguk, 2007.

[21] H. Woo, A. K. Mok, and C.-G. Lee. A Generic Framework for Monitoring Timing Constraints over Uncertain Events. In *Proc. of IEEE Real-Time Systems Symposium(RTSS)*, pages 435–444, December 2006.