

Article

Monitoring Threshold Functions over Distributed Data Streams with Node Dependent Constraints

Yaakov Malinovsky * and Jacob Kogan

Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD 21250, USA; E-Mail: kogan@umbc.edu

* Author to whom correspondence should be addressed; E-Mail: yaakovm@umbc.edu;
Tel.: +1-410-455-2968; Fax: +1-410-455-1066.

Received: 19 June 2012; in revised form: 8 September 2012 / Accepted: 11 September 2012 /
Published: 18 September 2012

Abstract: Monitoring data streams in a distributed system has attracted considerable interest in recent years. The task of feature selection (e.g., by monitoring the information gain of various features) requires a very high communication overhead when addressed using straightforward centralized algorithms. While most of the existing algorithms deal with monitoring simple aggregated values such as frequency of occurrence of stream items, motivated by recent contributions based on geometric ideas we present an alternative approach. The proposed approach enables monitoring values of an arbitrary threshold function over distributed data streams through stream dependent constraints applied separately on each stream. We report numerical experiments on a real-world data that detect instances where communication between nodes is required, and compare the approach and the results to those recently reported in the literature.

Keywords: data streams; distributed system; convex optimization; feedback; feature selection

1. Introduction

In many emerging applications one needs to process a continuous stream of data in real time. Sensor networks [1], network monitoring [2], and real-time analysis of financial data [3,4] are examples of such applications. Monitoring queries is a particular class of queries in the context of data streams. Previous

work in this area deals with monitoring simple aggregates [2], or term frequency occurrence in a set of distributed streams [5].

A general framework for efficient local algorithms monitoring l_2 norm of the data average of large networks of computers, wireless sensors, or mobile devices was introduced in [6], and further developed in [7]. The current contribution is motivated by results recently reported in [8,9] with focus on a special case of the general model considered in [7]. This special case can be briefly described as follows:

Let $S = \{s_1, \dots, s_n\}$ be a set of data streams collected at n nodes. Let $v_1(t), \dots, v_n(t)$ be d dimensional real time varying vectors derived from the streams. For a function $f : \mathbf{R}^d \rightarrow \mathbf{R}$ we would like to confirm the inequality

$$f\left(\frac{v_1(t) + \dots + v_n(t)}{n}\right) > 0 \quad (1)$$

while minimizing communication between the nodes. Monitoring inequality (1), or monitoring geometric location of the mean is a problem that can be addressed using a variety of different mathematical tools. A specific choice of a monitoring tool is up to the user. We note that the problem as stated above does not specify any particular tool, l_2 , or any other norm that is required to address it.

The problem was recently addressed in [10], where the approach proposed imposes equal constraints on each node. In addition to previously used l_2 norm (see, e.g., [6–9,11]) the paper provides theoretical framework for using a wide variety of convex functions, and, as an illustration, runs numerical experiments using l_2 , l_1 and l_∞ norms. In all numerical experiments reported in [10] an application of the same algorithm with l_1 norm generates superior results. This paper extends results in [10] in a machine learning direction—a constraint imposed on each node depends on the stream history at the node.

As a simple illustration of the problem considered in the paper we focus on two scalar functions $v_1(t)$ and $v_2(t)$, and the identity function f (*i.e.*, $f(x) = x$). We would like to guarantee the inequality

$$v(t) = \frac{v_1(t) + v_2(t)}{2} > 0$$

while keeping the nodes silent as much as possible. A possible strategy is to verify the initial inequality $v(t_0) = \frac{v_1(t_0) + v_2(t_0)}{2} > 0$ and to keep both nodes silent while

$$|v_i(t) - v_i(t_0)| < \delta = v(t_0), \quad t \geq t_0, \quad i = 1, 2$$

The first time t_1 when one of the functions, say $v_1(t)$, crosses the boundary of the local constraint, *i.e.*, $|v_1(t_1) - v_1(t_0)| \geq \delta$ the nodes communicate, the mean $v(t_1)$ is computed, the local constraint δ is updated and made available to the nodes, and nodes are kept silent as long as the inequalities hold.

$$|v_i(t) - v_i(t_1)| < \delta, \quad t \geq t_1, \quad i = 1, 2$$

The main contributions of this paper are listed next. We demonstrate that:

1. This approach works for a non-linear monitoring function f .
2. The results depend on the choice of a norm, and the numerical results reported show that l_2 is probably not the best norm when one aims to minimize communication between nodes. In addition to the numerical results presented we also provide a simple illustrative example that highlights this point (see Remark 4.2).

3. Selection of node dependent local constraints may decrease communication between the nodes.
4. The approach suggested in [10] and adopted in this paper paves the way to achieve further communication savings by clustering nodes, and monitoring cluster coordinators. Although this research direction is beyond the scope of this paper we address it briefly in Section 6.

In the next section we provide a text mining related example that leads to a non-linear threshold function f .

2. Text Mining Application

Let \mathbf{T} be a finite text collection (for example a collection of mail or news items). We denote the size of the set \mathbf{T} by $|\mathbf{T}|$. We will be concerned with two subsets of \mathbf{T} :

1. \mathbf{R} —the set of “relevant” texts (text not labeled as spam),
2. \mathbf{F} —the set of texts that contain a “feature” (word or term for example).

We denote complements of the sets by $\overline{\mathbf{R}}$, $\overline{\mathbf{F}}$ respectably (*i.e.*, $\mathbf{R} \cup \overline{\mathbf{R}} = \mathbf{F} \cup \overline{\mathbf{F}} = \mathbf{T}$), and consider the relative size of the four sets $\mathbf{F} \cap \overline{\mathbf{R}}$, $\mathbf{F} \cap \mathbf{R}$, $\overline{\mathbf{F}} \cap \overline{\mathbf{R}}$, and $\overline{\mathbf{F}} \cap \mathbf{R}$ as follows:

$$\begin{aligned} x_{11}(\mathbf{T}) &= \frac{|\mathbf{F} \cap \overline{\mathbf{R}}|}{|\mathbf{T}|}, & x_{12}(\mathbf{T}) &= \frac{|\mathbf{F} \cap \mathbf{R}|}{|\mathbf{T}|} \\ x_{21}(\mathbf{T}) &= \frac{|\overline{\mathbf{F}} \cap \overline{\mathbf{R}}|}{|\mathbf{T}|}, & x_{22}(\mathbf{T}) &= \frac{|\overline{\mathbf{F}} \cap \mathbf{R}|}{|\mathbf{T}|} \end{aligned} \quad (2)$$

Note that

$$0 \leq x_{ij} \leq 1, \text{ and } x_{11} + x_{12} + x_{21} + x_{22} = 1$$

The function f is defined on the simplex (*i.e.*, $x_{ij} \geq 0$, $\sum x_{ij} = 1$), and given by

$$f(x_{11}, x_{12}, x_{21}, x_{22}) = \sum_{i,j} x_{ij} \log \left(\frac{x_{ij}}{(x_{i1} + x_{i2})(x_{1j} + x_{2j})} \right) \quad (3)$$

where $\log x = \log_2 x$ throughout the paper. We next relate empirical version of information gain Equation (3) and the information gain (see e.g., [12]).

Let Y and X be random variable with known distributions

$$P(Y = y_i), \quad i = 1, \dots, n, \text{ and } P(X = x_j), \quad j = 1, \dots, m$$

Entropy of Y is defined by

$$H(Y) = - \sum_{i=1}^n P(Y = y_i) \log P(Y = y_i) \quad (4)$$

Entropy of Y conditional on $X = x$ denoted by $H(Y|X = x)$ is defined by

$$-\sum_{i=1}^n \frac{P(Y = y_i, X = x)}{P(X = x)} \log \frac{P(Y = y_i, X = x)}{P(X = x)} \quad (5)$$

Conditional entropy $H(Y|X)$ and information gain $IG(Y|X)$ are given by

$$\begin{aligned} H(Y|X) &= \sum_{j=1}^m P(X = x_j) H(Y|X = x_j) \\ &\quad \text{and} \\ IG(Y|X) &= H(Y) - H(Y|X) \end{aligned} \tag{6}$$

Information gain is symmetric, indeed

$$\begin{aligned} IG(Y|X) &= \\ &\sum_{i,j} P(Y = y_i, X = x_j) \log \frac{P(Y = y_i, X = x_j)}{P(X = x_j)} \\ &- \sum_i P(Y = y_i) \log P(Y = y_i) = \\ &\sum_{i,j} P(Y = y_i, X = x_j) \log \frac{P(Y = y_i, X = x_j)}{P(Y = y_i)P(X = x_j)} \\ &= IG(X|Y) \end{aligned}$$

Due to convexity of $g(x) = -\log x$, information gain is non-negative

$$\begin{aligned} IG(Y|X) &= \sum_{i,j} P(Y = y_i, X = x_j) g\left(\frac{P(Y = y_i)P(X = x_j)}{P(Y = y_i, X = x_j)}\right) \\ &\geq g\left(\sum_{i,j} P(Y = y_i, X = x_j) \frac{P(Y = y_i)P(X = x_j)}{P(Y = y_i, X = x_j)}\right) \\ &= g\left(\sum_{i,j} P(Y = y_i)P(X = x_j)\right) = -\log 1 = 0 \end{aligned}$$

It is easy to see that Equation (3) provides information gain for the “feature”.

As an example, we consider n agents installed on n different servers and a stream of texts arriving at the servers. Let $\mathbf{T}_h = \{\mathbf{t}_{h1}, \dots, \mathbf{t}_{hw}\}$ be the last w texts received at the h^{th} server, with $\mathbf{T} = \bigcup_{h=1}^n \mathbf{T}_h$. Note that

$$x_{ij}(\mathbf{T}) = \sum_{h=1}^n \frac{|\mathbf{T}_h|}{|\mathbf{T}|} x_{ij}(\mathbf{T}_h)$$

i.e., entries of the global contingency table $\{x_{ij}(\mathbf{T})\}$ are the average of the local contingency tables $\{x_{ij}(\mathbf{T}_h)\}$, $h = 1, \dots, n$.

For the given “feature” and a predefined positive threshold r we would like to verify the inequality

$$f(x_{11}(\mathbf{T}), x_{12}(\mathbf{T}), x_{21}(\mathbf{T}), x_{22}(\mathbf{T})) - r > 0$$

while minimizing communication between the servers. Note that Equation (3) is a nonlinear function. The case of a nonlinear monitoring function is different from that of linear one (in fact [8] calls the nonlinear monitoring function case “fundamentally different”). In the next section we demonstrate the difference, and describe an efficient way to handle the nonlinear case.

3. Non-Linear Threshold Function: An Example

We start with a slight modification of a simple one dimensional example presented in [8].

Example 3.1 Let $f(x) = x^2 - 9$, and v_i , $i = 1, 2$ are scalar values stored at two distinct nodes. Note that if $v_1 = -4$, and $v_2 = 4$, then

$$f(v_1) = f(v_2) = 7 > 0 \text{ and}$$

$$f\left(\frac{v_1 + v_2}{2}\right) = -9 < 0$$

If $v_1 = -2$, and $v_2 = 6$, then

$$f(v_1) = -5 < 0, \quad f(v_2) = 27 > 0 \text{ and}$$

$$f\left(\frac{v_1 + v_2}{2}\right) = -5 < 0$$

Finally, when $v_1 = 2$, and $v_2 = 6$ one has

$$\begin{aligned} f(v_1) &= -5 < 0, \quad f(v_2) = 27 > 0 \\ &\quad \text{and} \\ f\left(\frac{v_1 + v_2}{2}\right) &= 7 > 0 \end{aligned} \tag{7}$$

The simple illustrative example leads the authors of [8] to conclude that it is impossible to determine from the values of f at the nodes whether its value at the average is above the threshold or not. The remedy proposed is to consider the vectors $\mathbf{u}_j(t) = \mathbf{v}(t_i) + [\mathbf{v}_j(t) - \mathbf{v}_j(t_i)]$, $j = 1, \dots, n$, $t \geq t_i$ and to monitor the values of f on the convex hull $\text{conv } \{\mathbf{u}_1(t), \dots, \mathbf{u}_n(t)\}$ instead of the value of f at the average Equation (1). This strategy leads to sufficient conditions for Equation (1), and may be conservative.

The monitoring techniques for values of f on $\text{conv } \{\mathbf{u}_1(t), \dots, \mathbf{u}_n(t)\}$ without communication between the nodes are based on the following two observations:

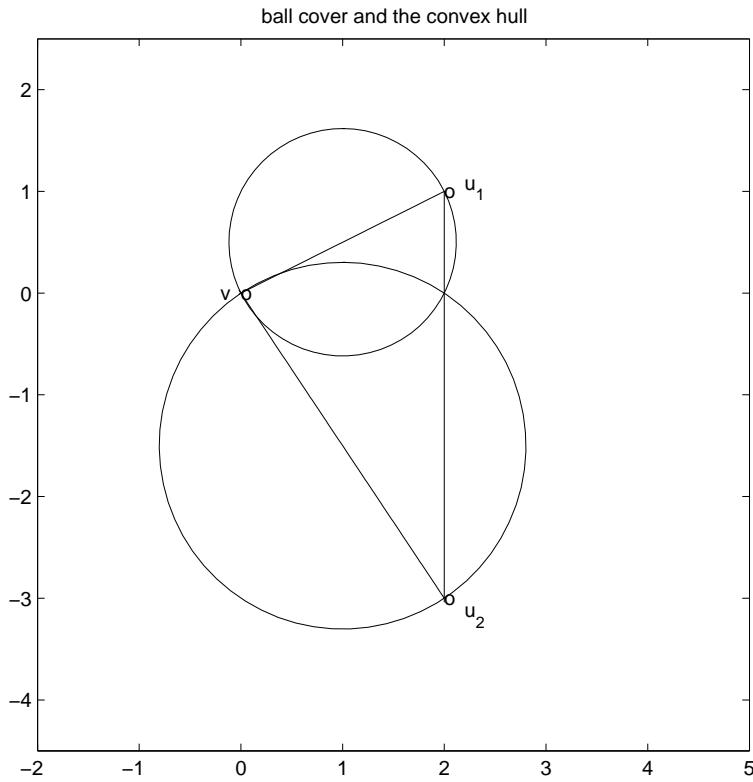
1. *Convexity property.* The mean $\mathbf{v}(t)$ is given by $\frac{\mathbf{v}_1(t) + \dots + \mathbf{v}_n(t)}{n} = \frac{\mathbf{u}_1(t) + \dots + \mathbf{u}_n(t)}{n}$, i.e., the mean $\mathbf{v}(t)$ is in the convex hull of $\{\mathbf{u}_1(t), \dots, \mathbf{u}_n(t)\}$, and $\mathbf{u}_j(t)$ is available to node j without much communication with other nodes.
2. If $B_2(\mathbf{x}, \mathbf{y})$ is an l_2 ball of radius $\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2$ centered at $\frac{\mathbf{x} + \mathbf{y}}{2}$, then

$$\text{conv } \{\mathbf{v}, \mathbf{u}_1, \dots, \mathbf{u}_n\} \subseteq \bigcup_{j=1}^n B_2(\mathbf{v}, \mathbf{u}_j) \tag{8}$$

(see Figure 1). Since each ball

$$B_2(\mathbf{v}(t_i), \mathbf{u}_j(t)), \quad t \geq t_i, \quad j = 1, \dots, n \tag{9}$$

can be monitored by node j with no communication with other nodes, Equation (8) allows to split monitoring of $\text{conv } \{\mathbf{v}(t_i), \mathbf{u}_1(t), \dots, \mathbf{u}_n(t)\}$, $t \geq t_i$ into n independent tasks executed by the n nodes separately and without communication.

Figure 1. ball cover.

While the inclusion Equation (8) holds when B_2 is substituted by B_p with $p \geq 2$ as we show later (see Remark 4.3) the inclusion fails when, for example, $p = 1$ (for experimental results obtained with different norms see Section 5).

In this paper we propose an alternative strategy that will be briefly explained next using Example 3.1, $f(x) = x^2 - 9$, and assignment provided by Equation (7). Let δ be a positive number. Consider two intervals of radius δ centered at $v_1 = 2$ and $v_2 = 6$, i.e., we are interested in the intervals

$$[2 - \delta, 2 + \delta], \text{ and } [6 - \delta, 6 + \delta]$$

If $v_1(t) \in [2 - \delta, 2 + \delta]$, $v_2(t) \in [6 - \delta, 6 + \delta]$, and δ is small, then the average $\frac{v_1(t) + v_2(t)}{2}$ is not far from $\frac{2+6}{2} = 4$, and $f\left(\frac{v_1(t) + v_2(t)}{2}\right)$ is not far from 7 (hence positive). In fact the sum of the intervals is the interval $[8 - 2\delta, 8 + 2\delta]$, and

$$4 - \delta \leq \frac{v_1(t) + v_2(t)}{2} \leq 4 + \delta$$

The “zero” points Z_f of f are -3 and 3 , and as soon as δ is large enough so that the interval $[4 - \delta, 4 + \delta]$ “hits” a point where f vanishes, communication between the nodes is required in order to verify Equation (1). In this particular example as long as $\delta \leq 1$, and, therefore,

$$\max\{|v_1(t) - v_1|, |v_2(t) - v_2|\} < \delta \quad (10)$$

no communication is required between the nodes.

The condition presented above is a sufficient condition that guarantees Equation (1). As any sufficient condition is, this condition can be conservative. In fact when the distance is provided by the l_2 norm, this sufficient condition is more conservative than the one provided by “ball monitoring” Equation (9) suggested in [8]. On the other hand, since only a scalar δ should be communicated to each node, the value of the updated mean $\mathbf{v}(t_i)$ should not be transmitted (hence communication savings are possible), and there is no need to compute the distance from the center of each ball $B_2(\mathbf{v}(t_i), \mathbf{u}_j(t))$, $j = 1, \dots, n$, $t > t_i$ to the zero set Z_f . For detailed comparison of results we refer the reader to [10].

We conclude the section by remarking that when inequality Equation (1) is reversed the same technique can be used to monitor the reversed inequality while minimizing communication between the nodes. We provide additional details in Section 5. In the next section we extend the above “monitoring with no communication” argument to the general vector setting. The approach suggested in the next section is motivated by an earlier research on robust stability of control systems (see e.g., [13]).

4. Convex Minimization Problem

In this section we state the monitoring problem as a convex minimization problem. For an appropriate analysis background we refer the interested reader to the classical monograph [14]. For the relevant convex analysis material see [15].

Consider the following optimization problem:

Problem 4.1 For a function $K : \mathbf{R}^{d+nd} \rightarrow \mathbf{R}$ concave with respect to the first d variables $\lambda_1, \dots, \lambda_d$ and convex with respect to the last nd variables x_1, \dots, x_{nd} , solve

$$\inf_{\mathbf{x}} \sup_{\boldsymbol{\lambda}} K(\boldsymbol{\lambda}, \mathbf{x}) \quad (11)$$

A solution for Problem 4.1 with appropriately selected $K(\boldsymbol{\lambda}, \mathbf{x})$ concludes the section.

The connection between Problem 4.1, and the monitoring problem is explained next. Let B be a $d \times nd$ matrix made of n blocks, where each block is the $d \times d$ identity matrix multiplied by $\frac{1}{n}$, so that for a set of n vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ in \mathbf{R}^d one has

$$B\mathbf{w} = \frac{\mathbf{v}_1 + \dots + \mathbf{v}_n}{n} \text{ where } \mathbf{w}^T = (\mathbf{v}_1^T, \dots, \mathbf{v}_n^T) \quad (12)$$

Assume that inequality Equation (1) holds for the vector \mathbf{w} , i.e., $f(B\mathbf{w}) > 0$. We are looking for a vector \mathbf{x} “nearest” to \mathbf{w} so that $f(B\mathbf{x}) = 0$, i.e., $B\mathbf{x} = \mathbf{z}$ for some $\mathbf{z} \in Z_f$ (where Z_f is the zero set of f , i.e., $Z_f = \{\mathbf{z} : f(\mathbf{z}) = 0\}$). We now fix $\mathbf{z} \in Z_f$ and denote the distance from \mathbf{w} to the set $\{\mathbf{x} : B\mathbf{x} = \mathbf{z}\}$ by $r(\mathbf{z})$. Note that for each \mathbf{y} inside the ball of radius $r(\mathbf{z})$ centered at \mathbf{w} , one has $B\mathbf{y} \neq \mathbf{z}$. If \mathbf{y} belongs to a ball of radius $r = \inf_{\mathbf{z} \in Z_f} r(\mathbf{z})$ centered at \mathbf{w} , then the inequality $f(B\mathbf{y}) > 0$ holds true.

Let $F(\mathbf{x})$ be a “norm” on \mathbf{R}^{nd} (specific functions F we run the numerical experiments with will be described later). The nearest “bad” vector problem described above is the following.

Problem 4.2 For $\mathbf{z} \in Z_f$ identify

$$r(\mathbf{z}) = \inf_{\mathbf{x}} F(\mathbf{x} - \mathbf{w}) \text{ subject to } B\mathbf{x} = \mathbf{z} \quad (13)$$

We note that Equation (13) is equivalent to $\inf_{\mathbf{x}} \left[\sup_{\boldsymbol{\lambda}} \left\{ F(\mathbf{x} - \mathbf{w}) - \boldsymbol{\lambda}^T (B\mathbf{x} - \mathbf{z}) \right\} \right]$. The function

$$K(\boldsymbol{\lambda}, \mathbf{x}) = F(\mathbf{x} - \mathbf{w}) - \boldsymbol{\lambda}^T (B\mathbf{x} - \mathbf{z})$$

is concave (actually linear) in $\boldsymbol{\lambda}$, and convex in \mathbf{x} . Hence (see e.g., [15])

$$\inf_{\mathbf{x}} \left[\sup_{\boldsymbol{\lambda}} \left\{ F(\mathbf{x} - \mathbf{w}) - \boldsymbol{\lambda}^T (B\mathbf{x} - \mathbf{z}) \right\} \right] = \sup_{\boldsymbol{\lambda}} \left[\inf_{\mathbf{x}} \left\{ F(\mathbf{x} - \mathbf{w}) - \boldsymbol{\lambda}^T (B\mathbf{x} - \mathbf{z}) \right\} \right]$$

The right hand side of the above equality can be conveniently written as follows

$$\begin{aligned} \sup_{\boldsymbol{\lambda}} \left[\inf_{\mathbf{x}} \left\{ F(\mathbf{x} - \mathbf{w}) - \boldsymbol{\lambda}^T (B\mathbf{x} - \mathbf{z}) \right\} \right] &= \\ \sup_{\boldsymbol{\lambda}} \left[\boldsymbol{\lambda}^T (\mathbf{z} - B\mathbf{w}) - \sup_{\mathbf{x}} \left\{ (B^T \boldsymbol{\lambda})^T (\mathbf{x} - \mathbf{w}) - F(\mathbf{x} - \mathbf{w}) \right\} \right] \end{aligned}$$

The conjugate $g^*(\mathbf{y})$ of a function $g(\mathbf{x})$ is defined by $g^*(\mathbf{y}) = \sup_{\mathbf{x}} \left\{ \mathbf{y}^T \mathbf{x} - g(\mathbf{x}) \right\}$ (see e.g., [15]). We note that

$$\sup_{\mathbf{x}} \left\{ (B^T \boldsymbol{\lambda})^T (\mathbf{x} - \mathbf{w}) - F(\mathbf{x} - \mathbf{w}) \right\} = F^*(B^T \boldsymbol{\lambda})$$

hence to compute

$$\sup_{\boldsymbol{\lambda}} \left[\inf_{\mathbf{x}} \left\{ F(\mathbf{x} - \mathbf{w}) - \boldsymbol{\lambda}^T (B\mathbf{x} - \mathbf{z}) \right\} \right]$$

one has to deal with

$$\sup_{\boldsymbol{\lambda}} \left[\boldsymbol{\lambda}^T (\mathbf{z} - B\mathbf{w}) - F^*(B^T \boldsymbol{\lambda}) \right]$$

For many functions g the conjugate g^* can be easily computed. Next we list conjugate functions for the most popular norms

1. $\|\mathbf{u}\|_\infty = \max_i |u_i|$
2. $\|\mathbf{u}\|_2 = \left(\sum_{i=1}^d u_i^2 \right)^{\frac{1}{2}}$
3. $\|\mathbf{u}\|_1 = \sum_{i=1}^d |u_i|$

$g(\mathbf{u})$	conjugate $g^*(\mathbf{y})$
$\ \mathbf{u}\ _\infty$	$+\infty$ if $\ \mathbf{y}\ _1 > 1$ 0 if $\ \mathbf{y}\ _1 \leq 1$
$\ \mathbf{u}\ _2$	$+\infty$ if $\ \mathbf{y}\ _2 > 1$ 0 if $\ \mathbf{y}\ _2 \leq 1$
$\ \mathbf{u}\ _1$	$+\infty$ if $\ \mathbf{y}\ _\infty > 1$ 0 if $\ \mathbf{y}\ _\infty \leq 1$

We note that some of the functions F we consider in this paper are different from l_p norms (see Table 1 for the list of the functions). We first select $F(\mathbf{x}) = \|\mathbf{x}\|_\infty$, and show below that in this case

$$r(\mathbf{z}) = \sup_{\boldsymbol{\lambda}} \left[\boldsymbol{\lambda}^T (\mathbf{z} - B\mathbf{w}) - F^*(B^T \boldsymbol{\lambda}) \right] = \|\mathbf{z} - B\mathbf{w}\|_\infty$$

Note that with the choice $F(\mathbf{x}) = \|\mathbf{x}\|_\infty$ the problem $\sup_{\boldsymbol{\lambda}} [\boldsymbol{\lambda}^T (\mathbf{z} - B\mathbf{w}) - F^*(B^T \boldsymbol{\lambda})]$ becomes

$$\sup_{\boldsymbol{\lambda}} \boldsymbol{\lambda}^T (\mathbf{z} - B\mathbf{w}) \text{ subject to } \|B^T \boldsymbol{\lambda}\|_1 \leq 1$$

Since $\|B^T \boldsymbol{\lambda}\|_1 = \|\boldsymbol{\lambda}\|_1$ the problem reduces to

$$\sup_{\boldsymbol{\lambda}} \boldsymbol{\lambda}^T (\mathbf{z} - B\mathbf{w}) \text{ subject to } \|\boldsymbol{\lambda}\|_1 \leq 1$$

The solution to this maximization problem is $\|\mathbf{z} - B\mathbf{w}\|_\infty$. Analogously, when

$$\mathbf{x}^T = (\mathbf{x}_1^T, \dots, \mathbf{x}_n^T), \quad \mathbf{y}^T = (\mathbf{y}_1^T, \dots, \mathbf{y}_n^T) \in \mathbf{R}^{nd}, \text{ and } F(\mathbf{x}) = \max_i \{\|\mathbf{x}_i\|_2\}$$

one has $F^*(\mathbf{y}) = \sup_{\mathbf{x}} \left(\sum_{i=1}^n \mathbf{y}_i^T \mathbf{x}_i - \max_i \{\|\mathbf{x}_i\|_2\} \right)$. Assuming $\max_i \{\|\mathbf{x}_i\|_2\} = 1$ one has to look at

$$\sup_{\{\mathbf{x} : \|\mathbf{x}_i\|_2 \leq 1\}} \sum_{i=1}^n \mathbf{y}_i^T \mathbf{x}_i - 1 = \sum_{i=1}^n \|\mathbf{y}_i\|_2 - 1$$

Hence

$$F^*(\mathbf{y}) = \begin{cases} +\infty & \text{if } \sum_{i=1}^n \|\mathbf{y}_i\|_2 > 1 \\ 0 & \text{if } \sum_{i=1}^n \|\mathbf{y}_i\|_2 \leq 1 \end{cases}$$

and $\|B^T \boldsymbol{\lambda}\|_2 = \frac{1}{n} n \|\boldsymbol{\lambda}\|_2 = \|\boldsymbol{\lambda}\|_2$. Finally the value for $r(\mathbf{z})$ is given by $\|\mathbf{z} - B\mathbf{w}\|_2$. When $F(\mathbf{x}) = \max_i \{\|\mathbf{x}_i\|_1\}$ one has $r(\mathbf{z}) = \|\mathbf{z} - B\mathbf{w}\|_\infty$. For clarity sake we collect the above results in Table 1.

Table 1. norm–ball radius correspondence for three different norms and fixed $\mathbf{w} \in \mathbf{R}^{nd}$.

$F(\mathbf{x})$	$r(\mathbf{z})$
$\max_i \{\ \mathbf{x}_i\ _1\}$	$\ \mathbf{z} - B\mathbf{w}\ _1$
$\max_i \{\ \mathbf{x}_i\ _2\}$	$\ \mathbf{z} - B\mathbf{w}\ _2$
$\ \mathbf{x}\ _\infty = \max_i \{\ \mathbf{x}_i\ _\infty\}$	$\ \mathbf{z} - B\mathbf{w}\ _\infty$

In the algorithm described below the norm is denoted just by $\|\cdot\|$ (numerical experiments presented in Section 5 are conducted with all three norms). The monitoring algorithm we propose is the following.

Algorithm 4.1 Threshold monitoring algorithm.

1. Set $i = 0$.
2. Until end of stream.
3. Set $\mathbf{v}_j = \mathbf{v}_j(t_i)$, $j = 1, \dots, n$ (i.e., remember “initial” values for the vectors).
4. Set $\delta = \inf_{\mathbf{z} \in Z_f} \|\mathbf{z} - B\mathbf{w}(t_i)\|$ (for definition of \mathbf{w} see Equation (12)).

5. Set $i = i + 1$.
6. If $\|\mathbf{v}_j - \mathbf{v}_j(t_i)\| < \delta$ for each $j = 1, \dots, n$
 - go to step 5
 - else
 - go to step 3

In what follows, we assume that transmission of a double precision real number amounts to broadcasting one message. The message computation is based on the assumption that all nodes are updated by a new text simultaneously. When mean update is required, a coordinator (root) requests and receives messages from the nodes.

We next count a number of messages that should be broadcast per one iteration if the local constraint δ is violated at least at one node. We shall denote the set of all nodes by \mathbf{N} , the set of nodes complying with the constraint by \mathbf{N}^C , and the set of nodes violating the constraint by \mathbf{N}^V (so that $\mathbf{N} = \mathbf{N}^C \cup \mathbf{N}^V$). The cardinality of the sets is denoted by $|\mathbf{N}|$, $|\mathbf{N}^C|$, and $|\mathbf{N}^V|$ respectively, so that $|\mathbf{N}| = |\mathbf{N}^C| + |\mathbf{N}^V|$. Assuming $|\mathbf{N}^V| > 0$ one has the following:

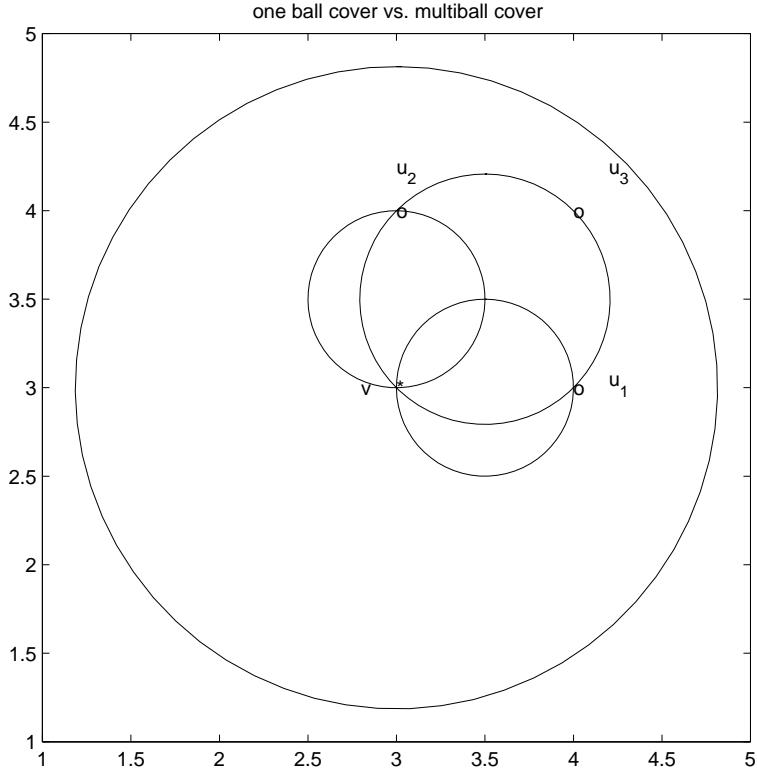
1. $|\mathbf{N}^V|$ violators transmit their scalar ID and new coordinates to the root ($(d + 1) \times |\mathbf{N}^V|$ messages).
2. the root sends scalar requests for new coordinates to the complying \mathbf{N}^C nodes ($|\mathbf{N}^C|$ messages).
3. the $|\mathbf{N}^C|$ complying nodes transmit new coordinates to the root ($d \times |\mathbf{N}^C|$ messages).
4. root updates itself, computes new distance δ to the surface, and sends δ to each node ($|\mathbf{N}|$ messages).

This leads to total of

$$(d + 2)|\mathbf{N}| \text{ messages per mean update.} \quad (14)$$

We conclude the section with three remarks. The first one compares conservatism of Algorithm 4.1 and the one suggested in [8]. The second one again compares the ball cover suggested in [8] and application of Algorithm 4.1 with l_1 norm. The last one shows by an example that Equation (8) fails when B_2 is substituted by B_1 . Significance of this negative result becomes clear in Section 5.

Remark 4.1 Let $\mathbf{v} = \frac{1}{n} \sum_{j=1}^n \mathbf{v}_j$, and $\mathbf{u}_j = \mathbf{v} + [\mathbf{v}_j(t_i) - \mathbf{v}_j]$. If the Step 6 inequality holds for each node, then each point of the ball centered at $\frac{\mathbf{v} + \mathbf{u}_j}{2}$ with radius $\left\| \frac{\mathbf{v} - \mathbf{u}_j}{2} \right\|_2$ is contained in the l_2 ball of radius δ centered at \mathbf{v} (see Figure 2). Hence the sufficient condition offered by Algorithm 4.1 is more conservative than the one suggested in [8].

Figure 2. conservative cover by a single l_2 ball.

Algorithm 4.1 can be executed with a variety of different norms, and, as we show next, l_2 might not be the best one when communication between the nodes should be minimized.

Remark 4.2 Let $n = d = 2$,

$$f(\mathbf{x}) = |x_1 - 1| + |x_2 - 1| = \|\mathbf{x} - \mathbf{e}\|_1$$

the distance is given by the l_1 norm, and the aim is to monitor the inequality $f(\mathbf{v}) - 1 > 0$. Let

$$\begin{aligned} \mathbf{v}_1(t_0) &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2(t_0) = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \\ \mathbf{v}_1(t_1) &= \begin{bmatrix} 1.9 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2(t_1) = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \end{aligned}$$

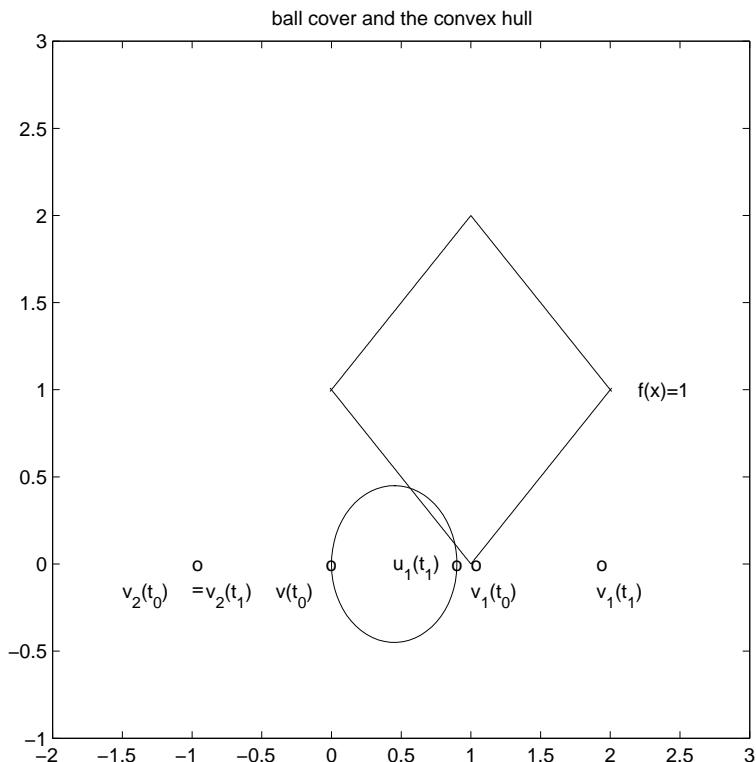
We first consider the “ball cover” construction suggested in [8]. With this data $\mathbf{v}(t_0) = 0$ with $f(\mathbf{v}(t_0)) = 2$, and $\mathbf{v}(t_1) = \begin{bmatrix} 0.45 \\ 0 \end{bmatrix}$ with $f(\mathbf{v}(t_1)) = 1.55$. At the same time $\mathbf{u}_1(t_1) = \mathbf{v}(t_0) + [\mathbf{v}_1(t_1) - \mathbf{v}_1(t_0)] = \begin{bmatrix} 0.9 \\ 0 \end{bmatrix}$. It is easy to see that the l_2 ball of radius $\left\| \frac{\mathbf{v}(t_0) - \mathbf{u}_1(t_1)}{2} \right\|_2$ centered at $\frac{\mathbf{v}(t_0) + \mathbf{u}_1(t_1)}{2}$ intersects the l_1 ball of radius 1 centered at $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ (see Figure 3). Hence the algorithm suggested in [8] requires nodes to communicate at time t_1 .

On the other hand the l_1 distance from $\mathbf{v}(t_0)$ to the set $\{\mathbf{x} : \|\mathbf{x} - \mathbf{e}\|_1 = 1\}$ is 1, and since

$$\|\mathbf{v}_1(t_1) - \mathbf{v}_1(t_0)\|_1 < 1, \text{ and } \|\mathbf{v}_2(t_1) - \mathbf{v}_2(t_0)\|_1 < 1$$

Algorithm 4.1 requires no communication between nodes at time t_1 . In this particular case the sufficient condition offered by Algorithm 4.1 is less conservative than the one suggested in [8].

Figure 3. l_2 ball cover requires communication.



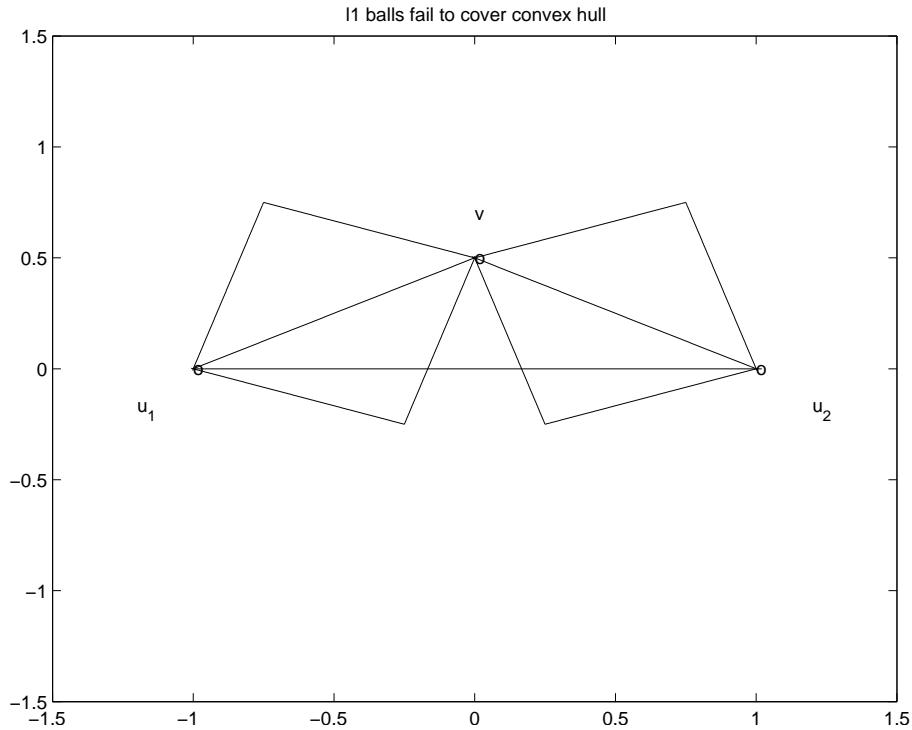
Remark 4.3 It is easy to see that inclusion Equation (8) fails when $B_1(\mathbf{x}, \mathbf{y})$ is an l_1 ball of radius $\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_1$ centered at $\frac{\mathbf{x} + \mathbf{y}}{2}$. Indeed, when, for example,

$$\mathbf{v} = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}, \quad \mathbf{u}_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad \mathbf{u}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

(see Figure 4) one has

$$\text{conv } \{\mathbf{v}, \mathbf{u}_1, \mathbf{u}_2\} \not\subset B_1(\mathbf{v}, \mathbf{u}_1) \cup B_1(\mathbf{v}, \mathbf{u}_2)$$

In the next section we apply Algorithm 4.1 to a real life data and report number of required mean computations.

Figure 4. failed cover by l_1 balls.

5. Experimental Results

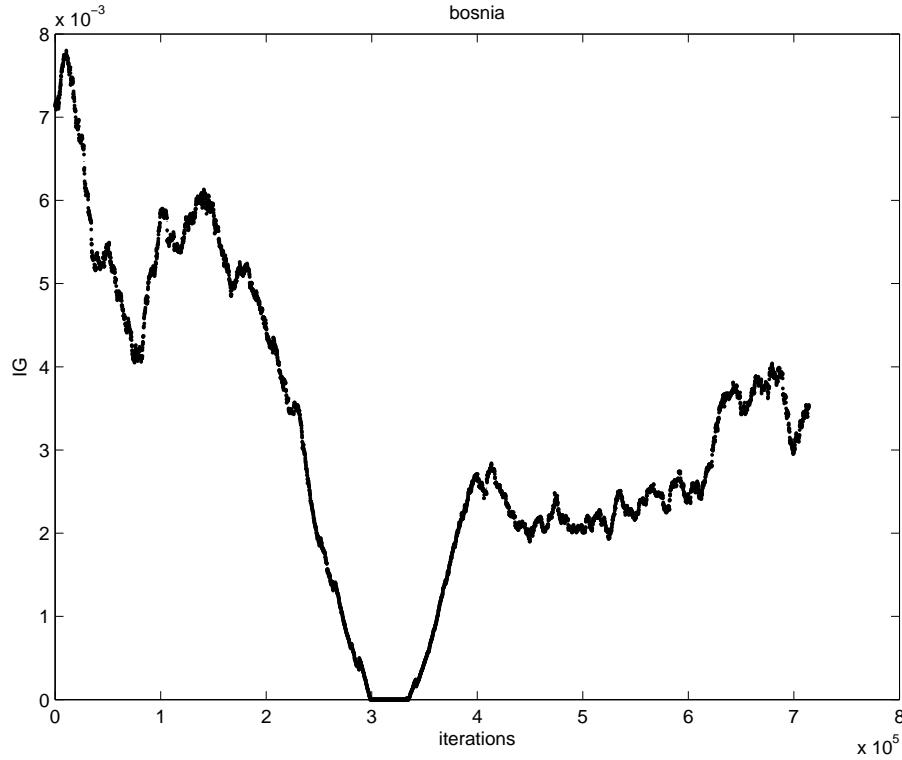
We apply Algorithm 4.1 to data streams generated from the Reuters Corpus RCV1–V2. The data is available from [16] and consists of 781,265 tokenized documents with DID (document ID) ranging from 2651 to 810596.

The methodology described below attempts to follow that presented in [8]. We simulate n streams by arranging the feature vectors in ascending order with respect to DID, and selecting feature vectors for the stream in the round robin fashion.

In the Reuters Corpus RCV1–V2 each document is labeled as belonging to one or more categories. We label a vector as “relevant” if it belongs to the “CORPORATE/INDUSTRIAL” (“CCAT”) category, and “spam” otherwise. Following [9] we focus on three features: “bosnia”, “ipo”, and “febru”. Each experiment was performed with 10 nodes, where each node holds a sliding window containing the last 6700 documents it received.

First we use 67,000 documents to generate initial sliding windows. The remaining 714,265 documents are used to generate data streams, hence the selected feature information gain is computed 714,265 times. Based on all the documents contained in the sliding window at each one of the 714,266 time instances, we compute and graph 714,266 information gain values for the feature “bosnia” (see Figure 5).

For the experiments described below the threshold value r is predefined, and the goal is to monitor the inequality $f(\mathbf{v}) - r > 0$ while minimizing communication between the nodes. From now on we shall assume simultaneous arrival of a new text at each node.

Figure 5. information gain values for the feature “bosnia”.

As new texts arrive, the local constraint (*i.e.*, inequalities $\|\mathbf{v}_j - \mathbf{v}_j(t_i)\| < \delta$, $j = 1, \dots, n$) at each node is verified. If at least one node violates the local constraint, the average $\mathbf{v}(t_i)$ is updated. Our numerical experiment with the feature “bosnia”, the l_2 norm, and the threshold $r = 0.0025$ (reported in [8] as the threshold for feature “bosnia” incurring the highest communication cost) shows overall 4006 computation of the mean vector. An application of Equation (14) yields 240,360 messages. We repeat this experiment with l_∞ , and l_1 norms. The results obtained and collected in Table 2 show that the smallest number of the mean updates is required for the l_1 norm.

Table 2. number of mean computations, messages, and crossings per norm for feature “bosnia” with threshold $r = 0.0025$.

Distance	Mean Comps	Messages	LL	LG	GL	GG
l_2	4006	240,360	959	2	2	3043
l_∞	3801	228,060	913	2	2	2884
l_1	3053	183,180	805	2	2	2244

Throughout the iterations the mean $\mathbf{v}(t_i)$ goes through a sequence of updates, and the values $f(\mathbf{v}(t_i))$ may be larger than, equal to, or less than the threshold r . We monitor the case $f(\mathbf{v}) \leq r$ the same way as that of $f(\mathbf{v}) > r$. In addition to the number of mean computations, we collect statistics concerning “crossings” (or lack of thereof), *i.e.*, number of instances when the location of the mean \mathbf{v} and its update \mathbf{v}' relative to the surface $\{\mathbf{x} : \mathbf{x} \in \mathbf{R}^d, f(\mathbf{x}) = r\}$ are either identical or different. Specifically over the monitoring period we denote by:

1. “LL” the number of instances when $f(\mathbf{v}) < r$ and $f(\mathbf{v}') < r$,
2. “LG” the number of instances when $f(\mathbf{v}) < r$ and $f(\mathbf{v}') > r$,
3. “GL” the number of instances when $f(\mathbf{v}) > r$ and $f(\mathbf{v}') < r$,
4. “GG” the number of instances when $f(\mathbf{v}) > r$ and $f(\mathbf{v}') > r$.

The number of “crossings” is reported in the last four columns of Table 2.

Note that variation of vectors $\mathbf{v}_i(t)$ does not have to be uniform. Taking on account distribution of signals at each node may lead to additional communication savings. We illustrate this statement by a simple example involving just two nodes. If, for example, there is a reason to believe that

$$2\|\mathbf{v}_1 - \mathbf{v}_1(t_i)\| \leq \|\mathbf{v}_2 - \mathbf{v}_2(t_i)\| \quad (15)$$

then the number of node violations may be reduced by imposing node dependent constraints

$$\|\mathbf{v}_1 - \mathbf{v}_1(t_i)\| < \delta_1 = \frac{2}{3}\delta, \text{ and } \|\mathbf{v}_2 - \mathbf{v}_2(t_i)\| < \delta_2 = \frac{4}{3}\delta$$

so that the faster varying signal at the second node enjoys larger “freedom” of change, while the inequality

$$\left\| \frac{\mathbf{v}_1 + \mathbf{v}_2}{2} - \frac{\mathbf{v}_1(t_i) + \mathbf{v}_2(t_i)}{2} \right\| < \frac{\delta_1 + \delta_2}{2} = \delta$$

holds true. Assignments of “weighted” local constraints requires information provided by Equation (15). With no additional assumptions about signal distribution, this information is not available. Unlike [11] we refrain from making assumptions regarding possible underlying data distributions, instead we estimate the weights as follows:

1. Start with the initial set of weights

$$w_1 = \dots = w_n = 1 \text{ (so that } \sum_{j=1}^n w_j = n) \quad (16)$$

2. As texts arrive at the next time instance t_{i+1} each node computes

$$W_j(t_{i+1}) = W_j(t_i) + \|\mathbf{v}_j(t_{i+1}) - \mathbf{v}_j(t_i)\|, \text{ with } W_j(t_0) = 1, j = 1, \dots, n$$

If at time t_i a local constraint is violated, then, in addition to $(d + 2)|\mathbf{N}|$ messages (see Equation (14)), each node j broadcasts $W_j(t_i)$ to the root, the root computes $W = \sum_{j=1}^n W_j(t_i)$, and transmits the updated weights

$$w_j = n \times \frac{W_j(t_i)}{W} \text{ (so that } \sum_{j=1}^n w_j = n)$$

back to node j .

Broadcasts of weights cause increase of total number of messages per iteration to

$$(d + 4)|\mathbf{N}| \quad (17)$$

With inequalities in Step 6 of Algorithm 4.1 substituted by $\|\mathbf{v}_j - \mathbf{v}_j(t_i)\| < \delta_j = w_j \delta$ the number of mean computations is reported in Table 3.

It is of interest to compare results presented in Table 3 with those reported, for example, in [9]. The comparison, however, is not an easy task. While [9] reports the threshold $r = 0.0025$ as the threshold value that incurred the highest communication cost, the paper leaves the concept of “communication cost” undefined (we define transmission of a double precision real number as a single “message”). In addition [9] provides a graph of “Messages vs. Threshold” only. It appears that the maximal value of “bosnia Messages vs. Threshold” graph is somewhere between 100,000 and 200,000.

Table 3. number of mean computations, messages, and crossings per norm for feature “bosnia” with threshold $r = 0.0025$, and stream dependent local constraint δ_j .

Distance	Mean Comps	Messages	LL	LG	GL	GG
l_2	2388	191,040	726	2	2	1658
l_∞	2217	177,360	658	2	2	1555
l_1	1846	147,680	611	2	2	1231

We repeat the experiments with “ipo” and “febru” and report the results in Tables 4 and 5 respectively. The results obtained with stream dependent local constraints is a significant improvement over those presented in [10]. Consistent with the results in [10] l_1 norm comes up as the norm that requires smallest number of mean updates in all reported experiments.

Table 4. number of mean computations, messages, and crossings per norm for feature “febru” with threshold $r = 0.0025$, and stream dependent local constraint δ_j .

Distance	Mean Comps	Messages
l_2	1491	119,280
l_∞	1388	111,040
l_1	1304	104,320

Table 5. number of mean computations, messages, and crossings per norm for feature “ipo” with threshold $r = 0.0025$, and stream dependent local constraint δ_j .

Distance	Mean Comps	Messages
l_2	7656	612,480
l_∞	7377	590,160
l_1	6309	504,720

6. Future Research Directions

In what follows we briefly outline a number of immediate research directions we plan to pursue.

The local constraints introduced in this paper depend on history of a data stream at each node, and variations $\|\mathbf{v}_j(t_{i+1}) - \mathbf{v}_j(t_i)\|$ over time contribute uniformly to local constraints. Attaching more weight to recent changes than to older ones may contribute to further improvement of monitoring process.

Table 6 (borrowed from [10]) shows that in about 75% of instances (3034 out of 4006) the mean $\mathbf{v}(t)$ is updated because of a single node violation. This observation naturally leads to the idea of clustering nodes, and independent monitoring of the node clusters equipped with a coordinator. The monitoring will become a two step procedure. At the first step node violations are checked in each node separately. If a node violates its local constraint, the corresponding cluster computes updated cluster coordinator. At the second step, violations of local constraints by coordinators are checked, and if at least one violation is detected the root is updated. Table 6 indicates that in most of the instances only one coordinator will be effected, and, since communication within cluster requires less messages, the two step procedure briefly described above has a potential to bring additional savings.

Table 6. number of nodes simultaneously violating local constraints. for feature “bosnia” with threshold $r = 0.0025$, and l_2 norm

nodes	violations
1	3034
2	620
3	162
4	70
5	38
6	26
7	34
8	17
9	5
10	0

We note that a standard clustering problem is often described as “... finding and describing cohesive or homogeneous chunks in data, the clusters” (see e.g., [17]). The monitoring data streams problem requires to assign to the same cluster i nodes \mathbf{N}_i so that the total change within cluster $\left\| \sum_{\mathbf{v} \in \mathbf{N}_i} \mathbf{v} - \mathbf{v}(t_j) \right\|$ is minimized, *i.e.*, nodes with **different** variations $\mathbf{v} - \mathbf{v}(t_j)$ that cancel out each other as much as possible should be assigned to the same cluster. Hence, unlike classical clustering procedures, one needs to combine “dissimilar” nodes together. This is a challenging new type of a difficult clustering problem.

Realistically, verification of inequality $f(\mathbf{x}) - r > 0$ should be conducted with an error margin (*i.e.*, the inequality $f(\mathbf{x}) - r - \epsilon > 0$ should be investigated, see [9]). A possible effect of an error margin on the required communication load is another direction of future research.

7. Conclusions

Monitoring streams over distributed systems is an important and challenging problem with a wide range of applications. In this paper we build on the approach for monitoring an arbitrary threshold functions suggested in [10], and introduce stream dependent local constraints that serve as a feedback monitoring mechanism. The obtained preliminary results indicate substantial improvement over those reported in [10], and demonstrate that monitoring with l_1 norm requires fewer updates than that with l_∞ or l_2 norm.

Acknowledgments

The authors thank anonymous reviewers whose valuable comments greatly enhanced exposition of the results. The work of the first author was supported in part by 2012 UMBC Summer Faculty Fellowship grant.

References

1. Madden, S.; Franklin, M.J. An Architecture for Queries Over Streaming Sensor Data. In *Proceedings of the ICDE 02*, San Jose, CA, 26 February–1 March 2002; pp. 555–556.
2. Dilman, M.; Raz, D. Efficient Reactive Monitoring. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communication Societies*, Anchorage, Alaska, 2001; pp. 1012–1019.
3. Zhu, Y.; Shasha, D. Statestream: Statistical Monitoring of Thousands of Data Streams in Real Time. In *Proceeding of the 28th international conference on Very Large Data Bases (VLDB)*, Hong Kong, China, 2002; pp. 358–369.
4. Yi, B.-K.; Sidiropoulos, N.; Johnson, T.; Jagadish, H.V.; Faloutsos, C.; Biliris, A. Online Datamining for Co-Evolving Time Sequences. In *Proceedings of ICDE 00*, IEEE Computer Society, San Diego, CA, 2000; pp. 13–22.
5. Manjhi, A.; Shkapenyuk, V.; Dhamdhere, K.; Olston, C. Finding (Recently) Frequent Items in Distributed Data Streams. In *Proceedings of the 21st International Conference on Data Engineering (ICDE 05)*, Tokyo, Japan, 2005; pp. 767–778.
6. Wolff, R.; Bhaduri, K.; Kargupta, H. Local L2-Thresholding Based Data Mining in Peer-to-Peer Systems. In *Proceedings of the SIAM International Conference on Data Mining (SDM 06)*, Bethesda, MD, USA, 2006; pp. 430–441.
7. Wolff, R.; Bhaduri, K.; Kargupta, H. A generic local algorithm with applications for data mining in large distributed systems. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 465–478.
8. Sharfman, I.; Schuster, A.; Keren, D. A geometric approach to monitoring threshold functions over distributed data streams. *ACM Trans. Database Syst.* **2007**, *23*, 23–29.
9. Sharfman, I.; Schuster, A.; Keren, D. A Geometric Approach to Monitoring Threshold Functions over Distributed Data Streams. In *Ubiquitous Knowledge Discovery*; May, M., Saitta, L., Eds.; Springer-Verlag: New York, NY, USA, 2010; pp. 163–186.

10. Kogan, J. Feature Selection over Distributed Data Streams through Convex Optimization. In *Proceedings of the Twelfth SIAM International Conference on Data Mining (SDM 2012)*, Anaheim, CA, USA, 2012; pp. 475–484.
11. Keren, D.; Sharfman, I.; Schuster, A.; Livne, A. Shape sensitive geometric monitoring. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 1520–1535.
12. Gray, R.M. *Entropy and Information Theory*; Springer–Verlag: New York, NY, USA, 1990.
13. Hinrichsen, D.; Pritchard, A.J. Real and Complex Stability Radii: A Survey. In *Control of Uncertain Systems*; Hinrichsen, D., Pritchard, A.J., Eds.; Birkhauser: Boston, MA, USA, 1990; pp. 119–162.
14. Rudin, W. *Principles of Mathematical Analysis*; McGraw-Hill: New York, NY, USA, 1976.
15. Rockafellar, R.T. *Convex Analysis*; Princeton University Press: Princeton, NJ, USA, 1970.
16. Bottou, L. Home Page. Available online: leon.bottou.org/projects/sgd (accessed on 14 September 2012).
17. Mirkin, B. *Clustering for Data Mining: A Data Recovery Approach*; Chapman & Hall/CRC: Boca Raton, FL, USA, 2005.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>.)