Feature Selection Over Distributed Data Streams

Jacob Kogan

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 a set of 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.

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]. This contribution is motivated by results recently reported in [6] where a more general type of monitoring query is described as follows:

J. Kogan (✉)
Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD 21250, USA
e-mail: kogan@umbc.edu

K. Yada (ed.), *Data Mining for Service*, Studies in Big Data 3,
DOI: 10.1007/978-3-642-45252-9_2,

Let $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$ be a set of data streams collected at m nodes. Let $\mathbf{v}_1(t), \ldots, \mathbf{v}_m(t)$ be n dimensional real time varying vectors derived from the streams. For a function $f : \mathbf{R}^n \to \mathbf{R}$ we would like to confirm the inequality

$$f\left(\frac{\mathbf{v}_1(t) + \cdots + \mathbf{v}_m(t)}{m}\right) > 0 \tag{1}$$

while minimizing communication between the nodes.

As a simple illustration consider the case of 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 check the initial inequality $v(t_0) = \dfrac{v_1(t_0) + v_2(t_0)}{2} > 0$ and to keep both nodes silent while

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

The first time t_1 when one of the functions, say $v_1(t)$, hits the boundary of the local constraint, i.e. $|v_1(t_1) - v_1(t_0)| = \delta$ the nodes communicate, the new 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

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

hold. This type of monitoring will work for a general model (1) with a linear threshold function $f(\mathbf{v}) = \mathbf{a}^T\mathbf{v} + b$. In the next section we provide a Text Mining related example that leads to a non linear threshold function f.

2 Text Mining Application

We first introduce some preliminaries. For $\mathbf{x} \in \mathbf{R}^n$ we denote $\left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}$ by $\|\mathbf{x}\|_2$. Throughout the text $\log x = \log_2 x$. Let Y and X be random variable with know distributions

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

Entropy of Y is given by

$$H(Y) = -\sum_{i=1}^{n} P(Y = y_i) \log P(Y = y_i). \tag{2}$$

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

$$H(Y|X = x) = -\sum_{i=1}^{n} \frac{P(Y = y_i, X = x)}{P(X = x)} \log \frac{P(Y = y_i, X = x)}{P(X = x)}. \tag{3}$$

Conditional Entropy $H(Y|X)$ and Information Gain $IG(Y|X)$ are given by

$$H(Y|X) = \sum_{j=1}^{m} P(X = x_j) H(Y|X = x_j), \text{ and } IG(Y|X) = H(Y) - H(Y|X). \tag{4}$$

Information Gain is symmetric, i.e.

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

Let $\mathbf{T}$ be a finite text collection (or collection of mail 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, term).

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

$$\begin{aligned} x_{11} = x_{11}(\mathbf{T}) &= \frac{|\mathbf{F} \cap \overline{\mathbf{R}}|}{|\mathbf{T}|}, \; x_{12} = x_{12}(\mathbf{T}) = \frac{|\mathbf{F} \cap \mathbf{R}|}{|\mathbf{T}|}, \\ x_{21} = x_{21}(\mathbf{T}) &= \frac{|\overline{\mathbf{F}} \cap \overline{\mathbf{R}}|}{|\mathbf{T}|}, \; x_{22} = x_{21}(\mathbf{T}) = \frac{|\overline{\mathbf{F}} \cap \mathbf{R}|}{|\mathbf{T}|}. \end{aligned} \tag{5}$$

Note that

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

The function $f : \mathbf{R}^4 \to \mathbf{R}^1$

$$f(x_{11}, x_{12}, x_{21}, x_{22}) = \sum_{i,j} x_{ij} \log\left(\frac{x_{ij}}{(x_{i1} + x_{i2})(x_{1j} + x_{2j})}\right), \tag{6}$$

defined on the simplex (i.e. $x_i \geq 0$, $\sum x_i = 1$), provides information gain for the "feature".

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

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

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

For the given "feature" and a predefined threshold r we would like to verify the inequality

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

while minimizing communication between the servers. Note that (6) is a non linear function. The case of a nonlinear monitoring function is different from that of a linear one (in fact [7] calls the nonlinear monitoring function case "fundamentally different"). In Sect. 3 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 [7].

Example 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

$$f(v_1) = f(v_2) = 7 > 0, \quad \text{but} \quad f\left(\frac{v_1 + v_2}{2}\right) = -9 < 0.$$

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

$$f(v_1) = -5 < 0, \ f(v_2) = 27 > 0, \quad \text{but} \quad f\left(\frac{v_1 + v_2}{2}\right) = -5 < 0.$$

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

$$f(v_1) = -5 < 0, \;\; f(v_2) = 27 > 0, \quad \text{but} \quad f\left(\frac{v_1 + v_2}{2}\right) = 7 > 0. \tag{7}$$

The simple example leads the authors of [7] 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

$$\mathbf{u}_j(t) = \mathbf{v}(t_i) + [\mathbf{v}_j(t) - \mathbf{v}_j(t_i)], \;\; j = 1, \ldots, m, \;\; t \geq t_i$$

and to monitor the values of f on the convex hull conv $\{\mathbf{u}_1(t), \ldots, \mathbf{u}_m(t)\}$ instead of the value of f at the average (1). This strategy leads to sufficient conditions for (1), and may be conservative.

The monitoring techniques for values of f on conv $\{\mathbf{u}_1(t), \ldots, \mathbf{u}_m(t)\}$ with no communication between the nodes are based on two observations:

1. *Convexity property*. The mean $\mathbf{v}(t) = \dfrac{\mathbf{v}_1(t) + \cdots + \mathbf{v}_m(t)}{m} = \dfrac{\mathbf{u}_1(t) + \cdots + \mathbf{u}_m(t)}{m}$, i.e., the mean $\mathbf{v}(t)$ belongs to the convex hull of $\{\mathbf{u}_1(t), \ldots, \mathbf{u}_m(t)\}$ 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 a ball of radius $\dfrac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2$ centered at $\dfrac{\mathbf{x} + \mathbf{y}}{2}$, then

$$\text{conv } \{\mathbf{v}, \mathbf{u}_1, \ldots, \mathbf{u}_m\} \subseteq \bigcup_{j=1}^{m} B_2(\mathbf{v}, \mathbf{u}_j). \tag{8}$$

The second observation allows to break the task of monitoring

$$\text{conv } \{\mathbf{v}(t_i), \mathbf{u}_1(t), \ldots, \mathbf{u}_m(t)\}$$

into separate monitoring of each ball

$$B_2(\mathbf{v}(t_i), \mathbf{u}_j(t)), \;\; t \geq t_i \tag{9}$$

executed by node j without communication with other nodes.

In this chapter we propose an alternative strategy that will be briefly explained next using Example 1, and assignment provided by (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

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

When δ is small, $v_1(t) \in [2 - \delta, 2 + \delta]$, and $v_2(t) \in [6 - \delta, 6 + \delta]$ the average $\dfrac{v_1(t) + v_2(t)}{2}$ is not far from $\dfrac{2 + 6}{2}$, 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. 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 (1). In this particular example the "large enough" $\delta = 1$, and no communication between the nodes is required as long as

$$\max\{|v_1(t) - v_1|, |v_2(t) - v_2|\} < 1. \tag{10}$$

The condition presented above is a sufficient condition that guarantees (1). As any sufficient condition is 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" (9) suggested in [7]. On the other hand 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 $B_2(\mathbf{v}(t_i), \mathbf{u}(t_i))$ to the zero set Z_f. For detailed comparison of results we refer the reader to Sect. 4.

We conclude the section remarking that when inequality (1) is reversed the same technique can be used to minimize communication between nodes until f vanishes. We provide additional details in Sect. 5. In the Sect. 4 we extend the above "monitoring with no communication argument" to the general vector setting.

4 Convex Minimization Problem

In this section we show that monitoring problem can be stated as the following optimization problem.

Problem 1. For a function $K : \mathbf{R}^{n+nm} \to \mathbf{R}$ concave with respect to the first n variables $\lambda_1, \ldots, \lambda_n$ and convex with respect to the last nm variables $x_1, \ldots, x_{nm}$ solve

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

A solution for Problem 1 with appropriately selected $K(\boldsymbol{\lambda}, \mathbf{x})$ concludes the section.

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

$$B\mathbf{w} = \frac{\mathbf{v}_1 + \cdots + \mathbf{v}_m}{m}, \quad \text{where} \quad \mathbf{w}^T = \left(\mathbf{v}_1^T, \ldots, \mathbf{v}_m^T\right).$$

Assume that inequality (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. $B\mathbf{x} = \mathbf{z} \in Z_f$

(where Z_f is the zero set of f, i.e. $Z_f = \{\mathbf{z} : f(\mathbf{z}) = 0\}$). If the distance $r(\mathbf{z})$ between such $\mathbf{x}$ and $\mathbf{w}$ can be identified, than 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 $\mathbf{R}^{nm}$ (the exact function we run the numerical experiments with will be described later). The nearest "bad" vector problem described above is the following.

Problem 2. For $\mathbf{z} \in Z_f$ identify

$$r(\mathbf{z}) = \inf_{\mathbf{x}} F(\mathbf{x} - \mathbf{w}) \quad \text{subject to} \quad B\mathbf{x} = \mathbf{z}. \tag{12}$$

We note that (12) 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. [8])

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

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

$$\begin{aligned} &\sup_{\boldsymbol{\lambda}} \left[\inf_{\mathbf{x}} \left\{ F(\mathbf{x} - \mathbf{w}) - \boldsymbol{\lambda}^T (B\mathbf{x} - \mathbf{z}) \right\} \right] \\ &= \sup_{\boldsymbol{\lambda}} \left[\boldsymbol{\lambda}^T (\mathbf{z} - B\mathbf{w}) - \sup_{\mathbf{x}} \left\{ \left(B^T \boldsymbol{\lambda} \right)^T (\mathbf{x} - \mathbf{w}) - F(\mathbf{x} - \mathbf{w}) \right\} \right]. \end{aligned} \tag{13}$$

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. [8]). We note that

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

and the right hand side of (13) becomes

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

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

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

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

We select $F(\mathbf{x}) = \|\mathbf{x}\|_\infty$, and show below that

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

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

$$\sup_{\boldsymbol{\lambda}} \boldsymbol{\lambda}^T (\mathbf{z} - B\mathbf{w}) \quad \text{subject to} \quad \left|\left|B^T\boldsymbol{\lambda}\right|\right|_1 \le 1.$$

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

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

The solution to this maximization problem is $||\mathbf{z} - B\mathbf{w}||_\infty$. Analogously, when $F(\mathbf{x}) = \max_i\{\|\mathbf{x}_i\|_2\}$ one has

$$F^*(\mathbf{y}) = \sup_{\mathbf{x}} \left(\sum_{i=1}^m \mathbf{y}_i^T \mathbf{x}_i - \max_i\{\|\mathbf{x}_i\|_2\}\right)$$

Assuming $\max_i\{\|\mathbf{x}_i\|_2\} = 1$ one has to look at

$$\sup_{\|\mathbf{x}_i\|_2 \le 1} \sum_{i=1}^m \mathbf{y}_i^T \mathbf{x}_i - 1 = \sum_{i=1}^m \|\mathbf{y}_i\|_2 = \|\mathbf{y}\|_2.$$

Hence

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

and $\left|\left|B^T\boldsymbol{\lambda}\right|\right|_2 = \frac{1}{m} m\|\boldsymbol{\lambda}\|_2 = \|\boldsymbol{\lambda}\|_2$ 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 $r(\mathbf{z}) = ||\mathbf{z} - B\mathbf{w}||_1$. For this reason in the algorithm described below the norm is denoted just by $\|\cdot\|$.

The monitoring algorithm we propose is the following.

Algorithm 1 *Threshold monitoring algorithm.*

1. *Set* $i = 0$.
2. *Until end of stream.*
3. *Compute* $B\mathbf{w}(t_i)$ *the mean of the vectors* $\mathbf{v}_1(t_i), \ldots, \mathbf{v}_m(t_i)$.
4. *Set* $\overline{\mathbf{v}}_j = \mathbf{v}_j(t_i)$, $j = 1, \ldots, m$ *(i.e. remember "initial" values for the vectors).*
5. *Set* $\delta = \inf_{\mathbf{z} \in Z_f} ||\mathbf{z} - B\mathbf{w}(t_i)||$.
6. *Set* $i = i + 1$.
7. *If* $\|\overline{\mathbf{v}}_j - \mathbf{v}_j(t_i)\| < \delta$ *for each* $j = 1, \ldots, m$

   go to step 6

   else

   go to step 3

In what follows we assume that transmission of a double precision real number amounts to broadcasting one message. Next we consider two possible text arrival scenarios. In both cases one node is designated as a coordinator, and we assume that the coordinator can update the mean $\mathbf{v}(t_i)$ if supplied with vectors $\mathbf{v}_j(t_i)$ by the other nodes.

1. If only one node is updated at each time t_i, then the inequality $\|\overline{\mathbf{v}}_j - \mathbf{v}_j(t_i)\| < \delta$ in Step 7 should be checked for this node only. Step 3 of the algorithm requires to compute the mean. Each violation of the inequality in Step 7 triggers Step 3 execution (mean update). Assuming node m is a coordinator and node $j \neq m$ violates the Step 7 inequality a straightforward mean update procedure requires the following communications:

   a. node j sends $\mathbf{v}_j(t_i)$ to the coordinator (n broadcast),
   b. the coordinator updates $\mathbf{v}(t_i)$, updates the local constraint δ, and sends updated local constraint δ to nodes $j = 1, \ldots, m-1$ ($m-1$ broadcasts).

Overall execution of Step 3 requires broadcasting of

$$n + m - 1 \tag{14}$$

messages (here m is the number of nodes, and n is the dimension of the data vectors). The number of messages reduces to $m - 1$ if $j = m$.

2. If all nodes are updated by a new text simultaneously we shall denote by I_k the number of time instances when k nodes, $1 \le k \le m$ report violation of local constraint in Step 7, and by MU the overall required number of mean updates over the stream life time, so that $\sum_{k=1}^{m} I_k = MU$. Violation of Step 7 inequality by k nodes $j_1, \ldots, j_k$ requires the following communication:
   a. The k nodes send $(j_1, \mathbf{v}_{j_1}(t_i), \ldots, j_k, \mathbf{v}_{j_k}(t_i))$ to the coordinator ($k(1 + n)$ messages).
   b. the coordinator updates $\mathbf{v}(t_i)$, updates the local constraint δ, and sends updated local constraint δ to nodes $j = 1, \ldots, m - 1$ ($m - 1$ broadcasts).

   The total number of messages to be broadcasted does not exceed

$$(1 + n) \sum_{k=1}^{m} k I_k + MU(m - 1). \tag{15}$$

We conclude the section with the following remarks.

Remark 1. If the Step 7 inequality holds for each node, then each point of the ball centered at

$$\frac{\mathbf{v}(t_i) + \mathbf{v}(t_i) + [\mathbf{v}_j(t_i) - \mathbf{v}_j(t)]}{2} \quad \text{with radius} \quad \left\| \frac{\mathbf{v}_j(t_i) - \mathbf{v}_j(t)}{2} \right\|_2$$

is contained in the l_2 ball of radius δ centered at $\mathbf{v}(t_i)$. Hence the sufficient condition offered by Algorithm 1 is **more** conservative than the one suggested in [7].

Remark 2. Let $n = m = 2$, $f(\mathbf{x}) = |x_1 - 1| + |x_2 - 1| = \|\mathbf{x} - \mathbf{e}\|_1$, the distance is given by the l_1 norm, and the aim is to monitor the inequality $f(\mathbf{v}) - 1 > 0$. Let

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

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 $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Hence in this particular case the sufficient condition offered by Algorithm 1 is **less** conservative than the one suggested in [7].

Remark 3. It is easy to see that inclusion (8) fails when $B(\mathbf{x}, \mathbf{y})$ is a l_1 ball of radius $\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_1$, centered at $\frac{\mathbf{x}+\mathbf{y}}{2}$.

In Sect. 5 we apply Algorithm 1 to a real life data and report number of required mean computations.

5 Experimental Results

We apply Algorithm 1 to data streams generated from the Reuters Corpus RCV1–V2. The data is available from http://leon.bottou.org/projects/sgd 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 [7]. 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 [6] 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 datastreams, hence the selected feature information gain is changing 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 Fig. 1). At each one of the experiment the threshold value r is predefined, and the goal is to monitor the inequality $f(\mathbf{v}) - r > 0$.

Next we assume that new texts arrive simultaneously at each node, and the local constraint at each node is verified. If at some iteration at least one of the local constrains is violated the average $\mathbf{v}(t)$ is recomputed. Our numerical experiment with the feature "bosnia", the l_2 norm, and the threshold $r = 0.0025$ (reported in [7] as the threshold for feature "bosnia" incurring the highest communication cost), shows overall 4006 computation of the average vector with the node violation distribution reported in Table 1. An application of (15) yields 65079 messages.

Assuming arrival of a new text at one node only at any given time yields 4890 mean computations (see Table 4). An application of formula (14) yields 68460 messages. In both cases the required number of messages is significantly lower than the required number of messages reported in ([7], Fig. 8).

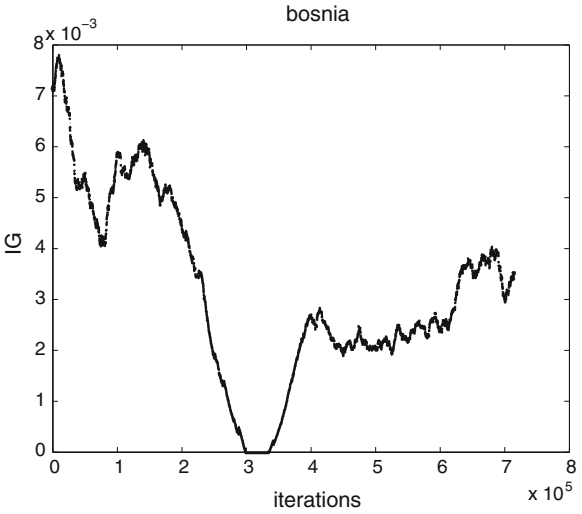

Fig. 1 Information gain: "bosnia"

Table 1 Number of local constraint violations simultaneously by k nodes for feature "bosnia" with threshold $r = 0.0025$, and l_2 norm

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

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. 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", i.e. updates $\mathbf{v}'$ of the mean $\mathbf{v}$ such that $f(\mathbf{v})$ and $f(\mathbf{v}')$ are on different sides of the separating surface $f(\mathbf{x}) = r$. The number of "crossings" is reported in the last four columns of the table. For example, the number of updates so that $f(\mathbf{v}) < r$ and $f(\mathbf{v}') < r$ is reported in column "LL" of Table 2.

Table 2 Number of average computations, and crossings for feature "bosnia" with threshold $r = 0.0025$

Distance	Total mean computations	LL	LG	GL	GG
l_2	4006	959	2	2	3043
l_∞	3801	913	2	2	2884
l_1	3053	805	2	2	2244

Table 3 Threshold, average computations, and crossings computed with l_1 norm for feature "bosnia"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	2122	207	1	1	1912
0.00125	3739	826	1	1	2910
0.00250	3675	2034	12	12	1616
0.00300	**5247**	3812	2	2	1430
0.00600	3255	3050	6	7	191

Table 4 Threshold, average computations, and crossings computed with l_2 norm for feature "bosnia"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	2694	249	1	1	2442
0.00125	5010	1120	1	1	3887
0.00250	4890	2674	12	12	2191
0.00300	**7629**	5681	4	4	1939
0.00600	4289	4003	8	9	268

From now on we assume that new texts arrive at the nodes at different times. At time t_0 the four dimensional vectors $\mathbf{v}_i(t_0) = \begin{bmatrix} x_{11}(\mathbf{T}_i) \\ x_{12}(\mathbf{T}_i) \\ x_{21}(\mathbf{T}_i) \\ x_{22}(\mathbf{T}_i) \end{bmatrix}$, the mean $\mathbf{v}(t_0)$, and the local constraint $\delta = \text{dist}\big(\mathbf{v}_1(t_0), \mathbf{Z}_{f-r}\big)$ are computed and made available to all the nodes (here $\mathbf{Z}_{f-r}$ is the zero set of the function $f(\mathbf{v})-r$, i.e. $\mathbf{Z}_{f-r} = \{\mathbf{v} : f(\mathbf{v}) = r\}$). The vectors $\overline{\mathbf{v}}_i = \mathbf{v}_i(t_0)$ are remembered at each node.

As a new text arrives at node 1 at time t_1 the vector $\mathbf{v}_1(t_1)$ is computed (while $\mathbf{v}_i(t_1) = \mathbf{v}_i(t_0)$, $i = 2, \ldots, m$ remain unchanged), and inequality $|\mathbf{v}_1(t_1) - \overline{\mathbf{v}}_1| < \delta$ is checked. If the inequality holds true, no updates of the mean $\mathbf{v}(t_0)$ and the local constraint δ are required, and the procedure is repeated for the nodes $i = 2, \ldots$ (see Algorithm 4.1). If the inequality fails the mean $\mathbf{v}(t_1)$ is updated by $\dfrac{\mathbf{v}_1(t_1) + \mathbf{v}_2(t_1) + \cdots + \mathbf{v}_m(t_1)}{m}$, the new local constraint $\delta = \text{dist}\,\big(\mathbf{v}(t_1), \mathbf{Z}_{f-r}\big)$ is computed, and made available to each node. Tables 3, 4 and 5 present the results obtained with l_1, l_2 and l_∞ norms respectively. In all three cases the largest number

Table 5 Threshold, average computations, and crossings computed with l_∞ norm for feature "bosnia"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	2368	210	1	1	2156
0.00125	4592	957	1	1	3633
0.00250	4737	2563	14	14	2146
0.00300	**7415**	5517	3	3	1892
0.00600	3954	3679	7	8	260

Table 6 Threshold, average computations, and crossings computed with l_1 norm for feature "ipo"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	3114	374	6	6	2727
0.00125	5899	1056	6	6	4830
0.00250	**15331**	4186	26	26	11092
0.00300	13712	8925	43	44	4699
0.00600	2820	2819	0	0	0

Table 7 Threshold, average computations, and crossings computed with l_2 norm for feature "ipo"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	3987	476	6	6	3498
0.00125	7774	1360	6	6	6401
0.00250	**21109**	6178	26	26	14878
0.00300	19923	13138	48	49	6687
0.00600	3679	3678	0	0	0

Table 8 Threshold, average computations, and crossings computed with l_∞ norm for feature "ipo"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	3703	470	6	6	3220
0.00125	7333	1323	6	6	5997
0.00250	19598	5984	25	25	13563
0.00300	**19653**	13264	49	50	6289
0.00600	3256	3255	0	0	0

of mean updates is required for the threshold value 0.00300. The results show that l_2 is probably not the most convenient norm to be used if the number of mean updates is to be minimized. It appears that computation performed with l_1 norm requires smallest number of mean updates for selected threshold values. The results of the experiments with items "ipo" are collected in Tables 6, 7 and 8. The "febru" relevant results are presented in Tables 9, 10, and 11. For all three features and five selected threshold values the l_1 norm requires the smallest number of mean updates.

Table 9 Threshold, average computations, and crossings computed with l_1 norm for feature "febru"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	3595	2041	16	16	1521
0.00125	**4196**	2419	37	37	1702
0.00250	2591	2216	6	6	362
0.00300	1683	1438	5	5	234
0.00600	506	505	0	0	0

Table 10 Threshold, average computations, and crossings computed with l_2 norm for feature "febru"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	4649	2803	19	19	1807
0.00125	**5360**	3164	37	37	2121
0.00250	3140	2698	7	7	427
0.00300	1941	1659	5	5	271
0.00600	547	546	0	0	0

Table 11 Threshold, average computations, and crossings computed with l_∞ norm for feature "febru"

Threshold	Total mean computations	LL	LG	GL	GG
0.00025	4426	2644	17	17	1747
0.00125	**5186**	3033	41	41	2070
0.00250	3044	2606	9	9	419
0.00300	1923	1634	5	5	278
0.00600	542	541	0	0	0

6 Conclusion

Monitoring streams over distributed systems is an important and challenging problem with a wide range of applications. In this short note we propose a new approach for monitoring an arbitrary threshold functions, and focus on the number of time instances when the global contingency table should be updated. The obtained preliminary results indicate that experiments with l_1 norm require fewer updates than those with l_∞ or l_2 norm. Identification of norms that are more appropriate for dealing with function f given by (6) is a future research direction.

Figure 1 inspection reveals that significant fraction of time the mean update may cause the local constraint δ to grow. A particular possible communication saving strategy is to keep the coordinator silent if the updated local constraint δ grows. Investigation of various balancing procedures for the coordinator (see, e.g., [6]) may lead to a significant reduction in communication cost. This is an additional research direction that will be pursuit. 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 [6]). A possible effect of an error margin on the required communication load is another direction of future research.

While the preliminary results appears to be promising additional research effort is needed to investigate effect of sliding window size, threshold and additional parameters of proposed algorithm performance.

Acknowledgments The author thanks Danny Keren for bringing the problem to his attention.

References

1. Madden, S., Franklin, M.J.: An architecture for queries over streaming sensor data. In: IEEE Computer Society, ICDE 02, p. 555. Washington, DC, USA (2002)
2. Dilman, M., Raz, D.: Efficient reactive monitoring. In Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communication Societies, pp. 1012–1019 (2001)
3. Yi, B.-K., Sidiropoulos, N., Johnson, T., Jagadish, H.V., Faloutsos, C., Biliris, A.: Online datamining for co-evolving time sequences. In: IEEE Computer Society, ICDE 00, p. 13. Washington, USA (2000)
4. Zhu, Y., Shasha, D.: Statestream: statistical monitoring of thousands of data streamsin real time. In: Very Large Data Base Endowment, pp. 358–369. (2002)
5. Manjhi, A., Shkapenyuk, V., Dhamdhere, K., Olston, C.: Finding (recently) frequent items in distributed data streams. In: IEEE Computer Society, ICDE 05, pp. 767–778. Los Alamitos, CA, USA (2005)
6. Sharfman, I., Schuster, A., Keren, D.: A geometric approach to monitoring threshold functions over distributed data streams. In: May, M., Saitta, L. (eds.) Ubiquitous Knowledge Discovery, pp. 163–186. Springer, New York (2010)
7. Sharfman, I., Schuster, A., Keren, D.: A geometric approach to monitoring threshold functions over distributed data streams. ACM Trans. Database Syst. **32**(4), 23:1–23:29 (2007)
8. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)

http://www.springer.com/978-3-642-45251-2

Data Mining for Service
Yada, K. (Ed.)
2014, VIII, 291 p. 109 illus., 12 illus. in color., Hardcover
ISBN: 978-3-642-45251-2