Deviation Estimation between Distributed Data Streams

Emmanuelle Anceaume, Yann Busnel

Deviation Estimation between Distributed Data Streams

Emmanuelle Anceaume
IRISA / CNRS
Rennes, France
Emmanuelle.Anceaume@irisa.fr

Yann Busnel
LINA / Université de Nantes
Nantes, France
Yann.Busnel@univ-nantes.fr

***Abstract*—The analysis of massive data streams is fundamental in many monitoring applications. In particular, for networks operators, it is a recurrent and crucial issue to determine whether huge data streams, received at their monitored devices, are correlated or not as it may reveal the presence of malicious activities in the network system. We propose a metric, called codeviation, that allows to evaluate the correlation between distributed streams. This metric is inspired from classical metric in statistics and probability theory, and as such allows us to understand how observed quantities change together, and in which proportion. We then propose to estimate the codeviation in the data stream model. In this model, functions are estimated on a huge sequence of data items, in an online fashion, and with a very small amount of memory with respect to both the size of the input stream and the values domain from which data items are drawn. We give upper and lower bounds on the quality of the codeviation, and provide both local and distributed algorithms that additively approximates the codeviation among n data streams by using $\mathcal{O}\left((1/\varepsilon)\log(1/\delta)\left(\log N + \log m\right)\right)$ bits of space for each of the n nodes, where N is the domain value from which data items are drawn, and m is the maximal stream's length. To the best of our knowledge, such a metric has never been proposed so far.**

***Index Terms*—Data stream model; correlation metric; distributed approximation algorithm; DDoS attacks.**

I. Introduction and background

Performance of many complex monitoring applications, including Internet monitoring applications, data mining, sensors networks, network intrusion/anomalies detection applications, depend on the detection of correlated events. For instance, detecting correlated network anomalies should drastically reduce the number of false positive or negative alerts that networks operators have to currently face when using network management tools such as SNMP or NetFlow. Indeed, to cope with the complexity and the amount of raw data, current network management tools analyze their input streams in isolation [1], [2]. Diagnosing flooding attacks through the detection of correlated flows should improve intrusions detection tools as proposed in [3], [4], [5]. In the same way, analyzing the effect of multivariate correlation for an early detection of Distributed Denial of Service (DDoS) is shown in [6]. The point is that, in all these monitoring applications, data streams arrive at nodes in a very high rate and may contain up to several billions of data items per day. Thus computing statistics with traditional methods is unpractical due to constraints on both available processing capacity, and memory. The problem tackled in this paper is the on-line estimation of data streams correlation. More precisely, we propose a distributed algorithm that approximates with guaranteed error bounds in a single pass the linear relation between massive distributed sequences of data.

Two main approaches exist to monitor in real time massive data streams. The first one consists in regularly sampling the input streams so that only a limited amount of data items is locally kept. This allows to exactly compute functions on these samples. However, accuracy of this computation with respect to the stream in its entirety fully depends on the volume of data items that has been sampled and their order in the stream. Furthermore, an adversary may easily take advantage of the sampling policy to hide its attacks among data items that are not sampled, or in a way that prevents its "malicious" data items from being correlated [7]. In contrast, the streaming approach consists in scanning each piece of data of the input stream on the fly, and in locally keeping only compact synopses or sketches that contain the most important information about these data. This approach enables to derive some data streams statistics with guaranteed error bounds without making any assumptions on the order in which data items are received at nodes. Most of the research done so far with this approach has focused on computing functions or statistics measures with error ε using poly$(1/\varepsilon, \log n)$ bits of space where n is the domain size of the data items. These include the computation of the number of different data items in a given stream [8], [9], [10], the frequency moments [11], the most frequent data items [11], [12], the entropy of the stream [13], [14], [15], or the information divergence over streams [16].

On the other hand, very few works have tackled the distributed streaming model, also called the functional monitoring problem [17], which combines features of both the streaming model and communication complexity models. As in the streaming model, the input data is read on the fly, and processed with a minimum workspace and time. In the communication complexity model, each node receives an input data stream, performs some local computation, and communicates only with a coordinator who wishes to continuously compute or estimate a given function of the union of all the input streams. The challenging issue in this model is for the coordinator to compute the given function by minimizing the number of communicated bits [17], [18], [19]. Cormode *et al.* [17] pioneer the formal study of functions in this model by focusing on the estimation of the first three frequency moments F_0, F_1 and F_2 [11]. Arackaparambil *et al.* [18] consider the empirical entropy estimation [11] and improve the work of Cormode by providing lower bounds on the frequency moments, and finally distributed algorithms for counting at any time t the number of items that have been received by a set of nodes from the inception of their streams have been proposed in [20], [21].

In this paper, we go a step further by studying the dispersion matrix of distributed streams. Specifically, we propose a novel metric that allows to approximate in real time the correlation between distributed and massive streams. This metric, called the sketch codeviation, allows us to quantify how observed data items change together, and in which proportion. As shown in [22], such a network-wide traffic monitoring tool should allow monitoring applications to get significant information on the traffic behaviour changes to subsequently inform more detailed detection tools on where DDoS attacks are currently active.

We give upper and lower bounds on the quality of this approximated metric with respect to the codeviation. As in [6], we use the codeviation analysis method, which is a statistical-based method that does not rely upon any knowledge of the nominal packet distribution. We then provide a distributed algorithm that additively approximates the codeviation among n data streams $\sigma_1, \ldots, \sigma_n$ by using $\mathcal{O}\left((1/\varepsilon)\log(1/\delta)\left(\log N + \log m\right)\right)$ bits of space for each of the n nodes, where N is the domain size from which items values are drawn, and m is the largest size of these data streams (more formally, $m = \max_{i\in[n]} \|X_{\sigma_i}\|_1$ where X_{σ_i} is the fingerprint vector representing the items frequency in stream σ_i). We guarantee that for any $0 < \delta < 1$, the maximal error of our estimation is bounded by $\varepsilon m/N$. To the best of our knowledge, such a work has never been done so far.

The remaining of the paper is organized as follows. First, Section II describes the computational model and some necessary background that makes the paper self-contained. Section III formalizes the sketch codeviation metric and studies its quality. Section IV presents the algorithm that computes the sketch codeviation between any two data streams, while Section V extends it to a distributed setting. Quality of both algorithms are analysed. Section VI presents some performance evaluation results. Finally, we conclude in Section VII.

II. Data Stream Model

A. Model

We present the computation model under which we analyze our algorithms and derive lower and upper bounds. We consider a set of n nodes $S_1, \ldots, S_n$ such that each node S_i receives a large sequence σ_{S_i} of data items or symbols. We assume that streams $\sigma_{S_1}, \ldots, \sigma_{S_n}$ do not necessarily have the same size, *i.e.*, some of the items present in one stream do not necessarily appear in others or their occurrence number may differ from one stream to another one. We also suppose that node S_i $(1 \leq i \leq n)$ does not know the length of its input stream. Items arrive regularly and quickly, and due to memory constraints (*i.e.*, nodes can locally store only a small amount of information with respect to the size of their input stream and perform simple operations on them), need to be processed sequentially and in an online manner. Nodes cannot communicate among each other. On the other hand, there exists a specific node, called the *coordinator* in the following, with which each node may communicate [17]. We assume that communication is instantaneous. We refer the reader to [23] for a detailed description of data streaming models and algorithms.

B. Preliminaries

We first present notations and background that make this paper self-contained. Let σ be a stream of data items that arrive sequentially. Each data item i is drawn from the universe $\Omega = \{1, 2, \ldots, N\}$, where N is very large. A natural approach to study a data stream σ of length m' is to model it as a fingerprint vector (or item frequency vector) over the universe Ω, given by $X = (x_1, x_2, \ldots, x_N)$ where x_i represents the number of occurrences of data item i in σ. Note that $0 \leq x_i \leq m'$. We have $\|X\|_1 = \sum_{i\in\Omega} x_i$, *i.e.*, $\|X\|_1$ is the norm of X. Thus $m' = \|X\|_1$.

1) Codeviation: In this paper, we focus on the computation of the deviation between any two streams using a space efficient algorithm with some error guarantee. The extension to a distributed environment $\sigma_1, \ldots, \sigma_n$ is studied in Section V. We propose a metric over fingerprint vectors of items, which is inspired from the classical covariance metric in statistics. Such a metric allows us to qualify the dependance or correlation between two quantities by comparing their variations. As will be shown in Section VI, this metric captures shifts in the network-wide traffic behavior when a DDoS attack is active. The codeviation between any two fingerprint vectors $X = (x_1, x_2, \ldots, x_N)$, and $Y = (y_1, y_2, \ldots, y_N)$ is the real number denoted $\mathrm{cod}(X, Y)$ defined by

$$\mathrm{cod}(X,Y) = \frac{1}{N}\sum_{i\in\Omega}(x_i - \overline{x})(y_i - \overline{y}) = \frac{1}{N}\sum_{i\in\Omega} x_i y_i - \overline{x}\,\overline{y} \quad (1)$$

$$\text{where } \overline{x} = \frac{1}{N}\sum_{i\in\Omega} x_i \text{ and } \overline{y} = \frac{1}{N}\sum_{i\in\Omega} y_i.$$

2) 2-universal Hash Functions: In the following, we use hash functions randomly picked from a 2-universal hash family. A collection H of hash functions $h : \{1, \ldots, M\} \to \{0, \ldots, M'\}$ is said to be *2-universal* if for every $h \in H$ and for every two different items $x, y \in [M]$, $\mathbb{P}\{h(x) = h(y)\} \leq \frac{1}{M'}$, which is exactly the probability of collision obtained if the hash function assigns truly random values to any $x \in [M]$.

3) Randomized (ε, δ)-additively-approximation Algorithm: A randomized algorithm $\mathcal{A}$ is said to be an (ε, δ)-additively-approximation of a function ϕ on σ if, for any sequence of items in the input stream σ, $\mathcal{A}$ outputs $\hat{\phi}$ such that $\mathbb{P}\{\mid \hat{\phi} - \phi \mid > \varepsilon\} < \delta$, where $\varepsilon, \delta > 0$ are given as parameters of the algorithm.

III. Sketch codeviation

As presented in the Introduction, we propose a statistic tool, named the sketch codeviation, which allows to approximate the codeviation between any two data streams using compact synopses or sketches. We then give bounds on the quality of this tool with respect to the computation of the codeviation applied on full streams.

Definition 1 (Sketch codeviation) *Let X and Y be any two fingerprint vectors of items, such that $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_N)$. Given a precision parameter k, we define the sketch codeviation between X and Y as*

$$\begin{aligned}\widehat{\mathrm{cod}}_k(X,Y) &= \min_{\rho\in\mathcal{P}_k(\Omega)} \mathrm{cod}\left(\widehat{X}_\rho, \widehat{Y}_\rho\right)\\ &= \min_{\rho\in\mathcal{P}_k(\Omega)} \left(\frac{1}{N}\sum_{a\in\rho}\widehat{X}_\rho(a)\widehat{Y}_\rho(a)\right.\\ &\qquad \left. - \left(\frac{1}{N}\sum_{a\in\rho}\widehat{X}_\rho(a)\right)\left(\frac{1}{N}\sum_{a\in\rho}\widehat{Y}_\rho(a)\right)\right)\end{aligned}$$

where $\forall a \in \rho, \widehat{X}_\rho(a) = \sum_{i\in a} x_i$, and $\mathcal{P}_k(\Omega)$ is a k-cell partition of Ω, i.e.*, the set of all the partitions of the set Ω into exactly k nonempty and mutually disjoint sets (or cells).*

Lemma 2 *Let $X = (x_1, \ldots, x_N)$, and $Y = (y_1, \ldots, y_N)$ be any two fingerprint vectors. We have*

$$\widehat{\mathrm{cod}}_N(X,Y) = \mathrm{cod}(X,Y)$$

Proof: It exists a unique partition ρ_N of N into exactly N nonempty and mutually disjoint sets, such that ρ_N is made of N singletons $\rho_N = \{\{1\},\{2\},\ldots,\{N\}\}$. Thus for any cell $a \in \rho_N$, there exists a unique $i \in \Omega$ such that $\widehat{X}_\rho(a) = x_i$. Thus, $\widehat{X}_\rho = X$ and $\widehat{Y}_\rho = Y$. ∎

Note that for $k > N$, it does not exist a partition of N into k nonempty parts. By convention, for $k > N$, $\widehat{\mathrm{cod}}_k(X,Y) = \widehat{\mathrm{cod}}_N(X,Y)$.

Proposition 3 *The sketch codeviation is a function of the codeviation. We have*

$$\widehat{\mathrm{cod}}_k(X,Y) = \mathrm{cod}(X,Y) + \mathcal{E}_k(X,Y)$$

$$\textit{where}\quad \mathcal{E}_k(X,Y) = \min_{\rho\in\mathcal{P}_k(\Omega)} \frac{1}{N}\sum_{a\in\rho}\sum_{i\in a}\sum_{j\in a\smallsetminus\{i\}} x_i y_j.$$

Proof: From Relation (1), we have

$$\begin{aligned}&\widehat{\mathrm{cod}}_k(X,Y)\\ &= \min_{\rho\in\mathcal{P}_k(\Omega)}\left(\left(\frac{1}{N}\sum_{a\in\rho}\widehat{X}_\rho(a)\widehat{Y}_\rho(a)\right)\right.\\ &\qquad\left. - \left(\frac{1}{N}\sum_{a\in\rho}\widehat{X}_\rho(a)\right)\left(\frac{1}{N}\sum_{a\in\rho}\widehat{Y}_\rho(a)\right)\right)\\ &= \min_{\rho\in\mathcal{P}_k(\Omega)}\left(\left(\frac{1}{N}\sum_{a\in\rho}\left(\sum_{i\in a}x_i\right)\left(\sum_{i\in a}y_i\right)\right)\right.\\ &\qquad\left. - \left(\frac{1}{N}\sum_{i\in\Omega}x_i\right)\left(\frac{1}{N}\sum_{j\in\Omega}y_j\right)\right)\\ &= \min_{\rho\in\mathcal{P}_k(\Omega)}\left(\left(\frac{1}{N}\sum_{a\in\rho}\left(\sum_{i\in a}\sum_{j\in a}x_i y_j\right)\right) - \overline{xy}\right)\\ &= \mathrm{cod}(X,Y) + \min_{\rho\in\mathcal{P}_k(\Omega)}\frac{1}{N}\sum_{a\in\rho}\sum_{i\in a}\sum_{j\in a\smallsetminus\{i\}}x_i y_j.\end{aligned}$$

which concludes the proof. ∎

The value $\mathcal{E}_k(X,Y)$ (which corresponds to the minimum sums over any partition ρ in $\mathcal{P}_k(\Omega)$) represents the *overestimation factor* of the sketch codeviation with respect to the codeviation.

A. Derivation of Lower Bounds on $\mathcal{E}_k(X,Y)$

We first show that if k is large enough, then the overestimation factor $\mathcal{E}_k(X,Y)$ is null, that is, the sketch codeviation matches exactly the codeviation.

Theorem 4 (Accuracy of the sketch codeviation) *Let X and Y be any two fingerprint vectors of items, such that $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_N)$. Then $\widehat{\mathrm{cod}}_k(X,Y) = \mathrm{cod}(X,Y)$ if*

$$\begin{aligned}k \geq |\,\mathrm{supp}(X)\cap\mathrm{supp}(Y)\,| &+ \mathbf{1}_{\mathrm{supp}(X)\smallsetminus\mathrm{supp}(Y)}\\ &+ \mathbf{1}_{\mathrm{supp}(Y)\smallsetminus\mathrm{supp}(X)}\end{aligned}$$

where $\mathrm{supp}(X)$*, respectively* $\mathrm{supp}(Y)$*, represents the support of distribution X, respectively Y (*i.e.*, the set of items in Ω that have a non null frequency $x_i \neq 0$, respectively $y_i \neq 0$, for $1 \leq i \leq N$), and notation $\mathbf{1}_A$ denotes the indicator function which is equal to 1 if the set A is not empty and 0 otherwise.*

Proof: Two cases are examined.

- **Case 1:**
  Let $k = |\,\mathrm{supp}(X)\cap\mathrm{supp}(Y)\,| + \mathbf{1}_{\mathrm{supp}(X)\smallsetminus\mathrm{supp}(Y)} + \mathbf{1}_{\mathrm{supp}(Y)\smallsetminus\mathrm{supp}(X)}$. We consider a partition $\overline{\rho} \in \mathcal{P}_k(\Omega)$ defined as follows

$$\begin{cases}\forall\ell\in\mathrm{supp}(X)\cap\mathrm{supp}(Y), \{\ell\}\in\overline{\rho}\\ \mathrm{supp}(X)\smallsetminus\mathrm{supp}(Y)\in\overline{\rho}\\ \mathrm{supp}(X)^{\complement}\in\overline{\rho}\end{cases}\tag{2}$$

Then from Relation (2) we have

$$\begin{cases} \forall \ell \in \operatorname{supp}(X) \cap \operatorname{supp}(Y), & \sum_{i \in \{\ell\}} \sum_{j \in \{\ell\} \smallsetminus \{i\}} x_i y_j = 0 \\ \forall \ell \in \operatorname{supp}(X) \smallsetminus \operatorname{supp}(Y), & y_\ell = 0 \\ \forall \ell \in \operatorname{supp}(X)^{\complement}, & x_\ell = 0. \end{cases}$$

Thus, $\sum_{a \in \overline{\rho}} \sum_{i \in a} \sum_{j \in a \smallsetminus \{i\}} x_i y_j = 0$. From Proposition (3), we get that $\widehat{\operatorname{cod}}_k(X,Y) = \operatorname{cod}(X,Y)$.

- **Case 2:**
For $k >| \operatorname{supp}(X) \cap \operatorname{supp}(Y) | + \mathbf{1}_{\operatorname{supp}(X) \smallsetminus \operatorname{supp}(Y)} + \mathbf{1}_{\operatorname{supp}(Y) \smallsetminus \operatorname{supp}(X)}$ (and $k < N$), it is always possible to split one of the two last cells of $\overline{\rho}$ as defined in Relation (2) with a singleton $\{\ell\}$ such that $x_\ell = 0$ or $y_\ell = 0$.

Both cases complete the proof. ∎

B. Derivation of Upper Bounds on $\mathcal{E}_k(X,Y)$

We have shown with Theorem 4 that the sketch codeviation matches exactly the codeviation if $k \geq| \operatorname{supp}(X) \cap \operatorname{supp}(Y) | + \mathbf{1}_{\operatorname{supp}(X) \smallsetminus \operatorname{supp}(Y)} + \mathbf{1}_{\operatorname{supp}(Y) \smallsetminus \operatorname{supp}(X)}$. In this section, we characterize the upper bound of the overestimation factor, *i.e.*, the error made with respect to the codeviation, when k is strictly less than this bound. To prevent problems of measurability, we restrict the classes of fingerprint vector under consideration. Specifically, given $m_{\mathcal{X}}$ and $m_{\mathcal{Y}}$ any positive integers, we define the two classes $\mathcal{X}$ and $\mathcal{Y}$ as $\mathcal{X} = \{X = (x_1, \ldots, x_N)$ such that $||X||_1 = m_{\mathcal{X}}\}$ and $\mathcal{Y} = \{Y = (y_1, \ldots, y_N)$ such that $||Y||_1 = m_{\mathcal{Y}}\}$. The following theorem derives the maximum value of the overestimation factor.

Theorem 5 (Upper bound of $\mathcal{E}_k(X,Y)$) *Let $k \geq 1$ be the precision parameter of the sketch codeviation. For any two fingerprint vectors $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, let $\mathcal{E}_k$ be the maximum value of the overestimation factor $\mathcal{E}_k(X,Y)$. Then, the following relation holds.*

$$\mathcal{E}_k = \max_{X \in \mathcal{X}, Y \in \mathcal{Y}} \mathcal{E}_k(X,Y) = \begin{cases} \dfrac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N} & \text{if } k = 1, \\ \dfrac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N}\left(\dfrac{1}{k} - \dfrac{1}{N}\right) & \text{if } k > 1. \end{cases}$$

Proof: The first part of the proof is directly derived from Lemma 6. Using Lemmata 7 and 8, we obtain the statement of the theorem. ∎

Lemma 6 *For any two fingerprint vectors $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, the maximum value $\mathcal{E}_1$ of the overestimation factor is exactly*

$$\mathcal{E}_1 = \max_{X \in \mathcal{X}, Y \in \mathcal{Y}} \mathcal{E}_1(X,Y) = \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N}.$$

Proof: $\forall X \in \mathcal{X}, \forall Y \in \mathcal{Y}$, we are looking for the maximal value of $\mathcal{E}_1(X,Y)$ under the following constraints:

$$\begin{cases} 0 \leq x_i \leq m_{\mathcal{X}} & \text{with } 1 \leq i \leq N, \\ 0 \leq y_i \leq m_{\mathcal{Y}} & \text{with } 1 \leq i \leq N, \\ \sum_{i=1}^N x_i = m_{\mathcal{X}}, \\ \sum_{i=1}^N y_i = m_{\mathcal{Y}}. \end{cases} \tag{3}$$

In order to relax one constraint, we set $x_N = m_{\mathcal{X}} - \sum_{i=1}^{N-1} x_i$. We rewrite $\mathcal{E}_1(X,Y)$ as a function f such that

$$\begin{aligned} & f(x_1, \ldots, x_{N-1}, y_1, \ldots, y_N) \\ & = \sum_{i=1}^{N-1} \sum_{j=1, j \neq i}^{N} x_i y_j + \left(m_{\mathcal{X}} - \sum_{i=1}^{N-1} x_i\right) \sum_{i=1}^{N-1} y_i. \end{aligned}$$

The function f is differentiable on its domain $[0..m_{\mathcal{X}}]^{N-1} \times [0..m_{\mathcal{Y}}]^N$. Thus we get

$$\begin{aligned} \frac{\mathrm{d}f}{\mathrm{d}x_i}(x_1, \ldots, x_{N-1}, y_1, \ldots, y_N) &= \sum_{j=1, j \neq i}^{N} y_j - \sum_{j=1}^{N-1} y_j \\ &= y_N - y_i. \end{aligned}$$

We need to consider the following two cases:

1) $y_N > y_i$. Function f is strictly increasing, and its maximum is reached for $x_i = m_{\mathcal{X}}$ (f is a Schur-convex function). By Relation 3, $\forall j \in \Omega \smallsetminus \{i\}, x_j = 0$.
2) $y_N \leq y_i$. Function f is decreasing, and its minimum is reached at $x_i = 0$.

By symmetry on Y, the maximum of $\mathcal{E}_1(X,Y)$ is reached for a distribution for which exactly one y_i is equal to $m_{\mathcal{Y}}$, and all the others y_j are equal to zero, which corresponds to the Dirac distribution. On the other hand, if the spike element of Y is the same as the one of X, then $\mathcal{E}_1(X,Y) = 0$, which is clearly not the maximum.

Thus, for all $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, the maximum $\mathcal{E}$ of the overestimation factor when $k = 1$ is reached for two Dirac distributions X^δ and Y^δ respectively centered in i and j with $i \neq j$, which leads to $\mathcal{E}_1 = \dfrac{1}{N} \sum_{i=1}^N \sum_{j=1, j \neq i}^N x_i^\delta \, y_j^\delta = \dfrac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N}$. ∎

We now show that for any $k > 1$, the maximum value of overestimation factor of the sketch codeviation between X and Y is obtained when both X and Y are uniform distributions.

Lemma 7 *Let X_U and Y_U be two uniform fingerprint vectors,* i.e., *$X_U = (x_1, \ldots, x_N)$ with $x_i = \frac{||X_U||_1}{N}$ for $1 \leq i \leq N$ and $Y_U = (y_1, \ldots, y_N)$ with $y_i = \frac{||Y_U||_1}{N}$ for $1 \leq i \leq N$. Then for any $k > 1$, the value of the overestimation factor is given by*

$$\mathcal{E}_k(X_U, Y_U) = \frac{||X_U||_1 ||Y_U||_1}{N}\left(\frac{1}{k} - \frac{1}{N}\right).$$

Proof: By definition, $\mathcal{E}_k(X_U, Y_U)$ represents for a given k the minimum overestimation factor for all k-cell partitions of Ω, and in particular for any regular partition for which all the k cells of the partition contain the same number $\frac{N}{k}$ of elements. In such a partition, all the k disjoint cells of the cross product matrix share the same value $\frac{||X_U||_1 ||Y_U||_1}{N^2}$. Therefore each cell a has the same weight equal to $\frac{||X_U||_1 ||Y_U||_1}{N^2}\left(\frac{N^2}{k^2} - \frac{N}{k}\right)$, leading to

$$\begin{aligned} \mathcal{E}_k(X_U, Y_U) &= \frac{k}{N} \frac{||X_U||_1 ||Y_U||_1}{N^2}\left(\frac{N^2}{k^2} - \frac{N}{k}\right) \\ &= \frac{||X_U||_1 ||Y_U||_1}{N}\left(\frac{1}{k} - \frac{1}{N}\right) \end{aligned}$$

which concludes the proof. ■

Lemma 8 *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be any two fingerprint vectors. Then the maximum value of the overestimation factor of the sketch codeviation when $k > 1$ is exactly*

$$\mathcal{E}_k = \max_{X\in\mathcal{X}, Y\in\mathcal{Y}} \mathcal{E}_k(X,Y) = \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N}\left(\frac{1}{k} - \frac{1}{N}\right).$$

Proof: Given $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ any two fingerprint vectors, let us denote $\mathcal{E}_k^{\rho}(X,Y) = \frac{1}{N}\sum_{a\in\rho}\sum_{i\in a}\sum_{j\in a\setminus\{i\}} x_i y_j$. Consider the partition $\overline{\rho} = \operatorname{argmin}_{\rho\in\mathcal{P}_k(\Omega)} \mathcal{E}_k^{\rho}(X,Y)$ with $k > 1$. We introduce the operator $\widetilde{\cdot}$ that operates on fingerprint vectors. This operator is defined as follows

- If it exists $a \in \overline{\rho}$ such that $\exists \ell, \ell' \in a$ with $y_\ell \geq y_{\ell'}$ and $x_{\ell'} > 0$, then operator $\widetilde{\cdot}$ is applied on the pair (ℓ, ℓ') of X so that we have $\begin{cases} \widetilde{x}_\ell = x_\ell + 1 \\ \widetilde{x}_{\ell'} = x_{\ell'} - 1 \end{cases}$.
- Otherwise, $\exists a, a' \in \overline{\rho}$ with $\exists \ell \in a, \exists \ell' \in a', x_\ell \geq x_{\ell'} > 0$. Then operator $\widetilde{\cdot}$ is applied on the pair (ℓ, ℓ') of X so that we have $\begin{cases} \widetilde{x}_\ell = x_\ell + 1 \\ \widetilde{x}_{\ell'} = x_{\ell'} - 1 \end{cases}$.
- Finally, X is kept unmodified for all the other items, *i.e.*, $\forall i \in \Omega \setminus \{\ell, \ell'\}, \widetilde{x}_i = x_i$.

It is clear that any fingerprint vectors can be constructed from the uniform one, using several iterations of this operator. Thus we split the proof into two parts. The first one supposes that both fingerprint vectors X and Y are uniform while the second part considers any two fingerprint vectors.

Case 1. Let X_U and Y_U be two uniform fingerprint vectors, *i.e.*, $X_U = (x_1, \ldots, x_N)$ with $x_i = \frac{||X_U||_1}{N}$ for $1 \leq i \leq N$ and $Y_U = (y_1, \ldots, y_N)$ with $y_i = \frac{||Y_U||_1}{N}$ for $1 \leq i \leq N$.

We split the analysis into two sub-cases: the class of partitions in which x_ℓ and $x_{\ell'}$ belong to the same cell a of a given k-partition ρ, and the class of partitions in which they are located into two separated cells a and a'. Suppose first that the $\widetilde{\cdot}$ operator is applied on X_U. Then the overestimation factor is given by

$$\mathcal{E}_k(\widetilde{X}_U, Y_U) = \min(E, E') \tag{4}$$

$$\text{with } \begin{cases} E = \displaystyle\min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a \in \rho, \ell, \ell' \in a}} \mathcal{E}_k^{\rho}(\widetilde{X}_U, Y_U) \\ E' = \displaystyle\min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a, a' \in \rho, a \neq a' \\ \wedge \ell \in a \wedge \ell' \in a'}} \mathcal{E}_k^{\rho}(\widetilde{X}_U, Y_U). \end{cases}$$

Let us consider the first term E. We have

$$\begin{aligned}
E &= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a \in \rho, \ell, \ell' \in a}} \Bigg(\sum_{b\in\rho\setminus\{a\}} \sum_{i\in b} \sum_{j\in b\setminus\{i\}} \widetilde{x}_i y_j \\
&\qquad + \sum_{i\in a} \sum_{j\in a\setminus\{i\}} \widetilde{x}_i y_j \Bigg) \\
&= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a \in \rho, \ell, \ell' \in a}} \Bigg(\sum_{b\in\rho\setminus\{a\}} \sum_{i\in b} \sum_{j\in b\setminus\{i\}} \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N^2} \\
&\qquad + \sum_{i\in a\setminus\{\ell,\ell'\}} \sum_{j\in a\setminus\{i\}} \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N^2} \\
&\qquad + \sum_{j\in a\setminus\{\ell\}} \left(\frac{m_{\mathcal{X}}}{N} + 1\right) \frac{m_{\mathcal{Y}}}{N} \\
&\qquad + \sum_{j\in a\setminus\{\ell'\}} \left(\frac{m_{\mathcal{X}}}{N} - 1\right) \frac{m_{\mathcal{Y}}}{N} \Bigg) \\
&= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a \in \rho, \ell, \ell' \in a}} \left(\mathcal{E}_k^{\rho}(X_U, Y_U)\right)
\end{aligned}$$

According to the second term E', we have

$$\begin{aligned}
E' &= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a, a' \in \rho, a \neq a' \\ \wedge \ell \in a \wedge \ell' \in a'}} \Bigg(\sum_{\substack{b \in \rho \\ \setminus\{a, a'\}}} \sum_{i\in b} \sum_{\substack{j \in b \\ \setminus\{i\}}} \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N^2} \\
&\qquad + \sum_{i\in a\setminus\{\ell\}} \sum_{j\in a\setminus\{i\}} \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N^2} \\
&\qquad + \sum_{i\in a'\setminus\{\ell'\}} \sum_{j\in a'\setminus\{i\}} \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N^2} \\
&\qquad + \sum_{j\in a\setminus\{\ell\}} \left(\frac{m_{\mathcal{X}}}{N} + 1\right) \frac{m_{\mathcal{Y}}}{N} \\
&\qquad + \sum_{j\in a'\setminus\{\ell'\}} \left(\frac{m_{\mathcal{X}}}{N} - 1\right) \frac{m_{\mathcal{Y}}}{N} \Bigg) \\
&= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a, a' \in \rho, a \neq a' \\ \wedge \ell \in a \wedge \ell' \in a'}} \left(\mathcal{E}_k^{\rho}(X_U, Y_U) + \frac{m_{\mathcal{Y}}}{N}\left(|a| - |a'|\right) \right)
\end{aligned}$$

Thus, $\mathcal{E}_k(\widetilde{X}_U, Y_U) \leq \mathcal{E}_k(X_U, Y_U)$. By symmetry, we have $\mathcal{E}_k(X_U, \widetilde{Y}_U) \leq \mathcal{E}_k(X_U, Y_U)$.

Case 2. In the rest of the proof, we show that for any X and Y, we have $\mathcal{E}_k(\widetilde{X}, Y) \leq \mathcal{E}_k(X, Y)$. Again, we split the proof into two sub-cases according to Relation 4. We get for the first term,

$$\begin{aligned}
&\min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a \in \rho, \ell, \ell' \in a}} \mathcal{E}_k^{\rho}(\widetilde{X}, Y) \\
&= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a \in \rho, \ell, \ell' \in a}} \left(\mathcal{E}_k^{\rho}(X, Y) + \sum_{j\in a\setminus\{\ell\}} y_j - \sum_{j\in a\setminus\{\ell'\}} y_j \right) \\
&= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a \in \rho, \ell, \ell' \in a}} \left(\mathcal{E}_k^{\rho}(X, Y) + y_{\ell'} - y_\ell \right).
\end{aligned}$$

For the second term, we have

$$\min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a,\, a' \in \rho,\, a \neq a' \\ \wedge \ell \in a \wedge \ell' \in a'}} \mathcal{E}_k^{\rho}(\widetilde{X}, Y)$$

$$= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a,\, a' \in \rho,\, a \neq a' \\ \wedge \ell \in a \wedge \ell' \in a'}} \left(\mathcal{E}_k^{\rho}(X, Y) + \sum_{j \in a \setminus \{\ell\}} y_j - \sum_{j \in a' \setminus \{\ell'\}} y_j \right).$$

By definition of the operator, if it exists $a \in \overline{\rho}$ such that $\exists \ell, \ell' \in a$, then $y_\ell \geq y_{\ell'}$ and so $\mathcal{E}_k^{\overline{\rho}}(\widetilde{X}, Y) \leq \mathcal{E}_k^{\overline{\rho}}(X, Y)$. Otherwise, ℓ and ℓ' are in two separated cells of $\overline{\rho}$, implying that $x_\ell \geq x_{\ell'}$. We then have $\sum_{j \in a \setminus \{\ell\}} y_j \leq \sum_{j \in a' \setminus \{\ell'\}} y_j$. Indeed, suppose that by contradiction

$$x_\ell \sum_{j \in a' \setminus \{\ell'\}} y_j + x_{\ell'} \sum_{j \in a \setminus \{\ell\}} y_j < \quad x_\ell \sum_{j \in a \setminus \{\ell\}} y_j + x_{\ell'} \sum_{j \in a' \setminus \{\ell'\}} y_j.$$

Let $\overline{\rho}'$ be the partition corresponding to the partition $\overline{\rho}$ in which ℓ and ℓ' have been swapped. Then we obtain $\mathcal{E}_k^{\overline{\rho}'}(X, Y) < \mathcal{E}_k^{\overline{\rho}}(X, Y)$, which is impossible by assumption on $\overline{\rho}$. Thus, in both cases we have $\mathcal{E}_k(\widetilde{X}, Y) \leq \mathcal{E}_k^{\overline{\rho}}(\widetilde{X}, Y) \leq \mathcal{E}_k^{\overline{\rho}}(X, Y) = \mathcal{E}_k(X, Y)$. By symmetry, we also have $\mathcal{E}_k(X, \widetilde{Y}) \leq \mathcal{E}_k(X, Y)$.

Thus we have shown that the maximum of any overestimation factor is reached for the uniform fingerprint vector. Lemma 7 concludes the proof. ■

So far, we have demonstrated that for any $k \geq 1$, the maximum value $\mathcal{E}_k$ of the overestimation factor of the sketch codeviation is less than or equal to $m_{\mathcal{X}} m_{\mathcal{Y}}/N$. We finally show that, given X and Y, the overestimation factor $\mathcal{E}_k(X, Y)$ is a decreasing function in k.

Lemma 9 *Let X and Y be any two fingerprint vectors. We have:*

$$\mathcal{E}_1(X, Y) \geq \mathcal{E}_2(X, Y) \geq \ldots \geq \mathcal{E}_k(X, Y) \geq \ldots \geq \mathcal{E}_N(X, Y).$$

Proof:

- **Case** $k = 1$. By assumption, $|\mathcal{P}_1(\Omega)| = 1$, *i.e.*, there exists a single partition which is the set Ω itself. Thus we directly have

$$\mathcal{E}_1(X, Y) = \frac{1}{N} \sum_{i \in \Omega} \sum_{j \in \Omega \setminus \{i\}} x_i y_j. \tag{5}$$

- **Case** $k = 2$. For any partition $\{a_1, a_2\} \in \mathcal{P}_2(\Omega)$, we have

$$\begin{aligned} &\mathcal{E}_1(X, Y) \\ &= \frac{1}{N} \left(\sum_{i \in a_1} \sum_{j \in a_1 \setminus \{i\}} x_i y_j + \sum_{i \in a_1} \sum_{j \in a_2} x_i y_j \right. \\ &\qquad \left. + \sum_{i \in a_2} \sum_{j \in a_1} x_i y_j + \sum_{i \in a_2} \sum_{j \in a_2 \setminus \{i\}} x_i y_j \right) \\ &= \mathcal{E}_2^{\rho}(X, Y) + \frac{1}{N} \left(\sum_{i \in a_1} \sum_{j \in a_2} x_i y_j + \sum_{i \in a_2} \sum_{j \in a_1} x_i y_j \right) \\ &\geq \mathcal{E}_2(X, Y). \end{aligned}$$

- **Case** $2 < k < N$. Let $\overline{\rho} = \operatorname{argmin}_{\rho \in \mathcal{P}_k(\Omega)} \mathcal{E}_k^{\rho}(X, Y)$, *i.e.*, partition $\overline{\rho}$ minimizes the overestimation factor for a given k. Then, there exists a partition $\rho' \in \mathcal{P}_{k+1}(\Omega)$ that can be obtained by splitting a cell of $\overline{\rho}$ in two cells, and constructed as follows

$$\begin{cases} \exists a_0 \in \overline{\rho}, \quad \exists a_1, a_2 \in \rho', \text{ such that } a_0 = a_1 \cup a_2 \\ \forall a \in \overline{\rho}, \quad a \neq a_0 \Rightarrow \exists a' \in \rho', \text{ such that } a = a'. \end{cases}$$

By using an argument similar to the previous one, we have

$$\begin{aligned} &\mathcal{E}_k(X, Y) \\ &= \mathcal{E}_{k+1}^{\rho'}(X, Y) + \frac{1}{N} \left(\sum_{i \in a_1} \sum_{j \in a_2} x_i y_j + \sum_{i \in a_2} \sum_{j \in a_1} x_i y_j \right) \\ &\geq \mathcal{E}_{k+1}(X, Y). \end{aligned}$$

Lemma 2 concludes the proof. ■

IV. Approximation Algorithm

In this section, we propose a one-pass algorithm that computes the sketch codeviation between any two large input streams. By definition of the metric (*cf.* Definition 1), we need to generate all the possible k-cell partitions. The number of these partitions follows the Stirling numbers of the second kind, which is equal to $S(N, k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^N$. Therefore, $S(N, k)$ grows exponentially with N. We show in the following that generating $t = \lceil \log(1/\delta) \rceil$ random k-cell partitions, where δ is the probability of error of our randomized algorithm, is sufficient to guarantee good overall performance of the sketch codeviation metric.

Our algorithm is inspired from the Count-Min Sketch algorithm proposed by Cormode and Muthukrishnan [24]. Specifically, the Count-Min algorithm is an (ε, δ)-approximation algorithm that solves the *frequency-estimation* problem. For any item v in the input stream σ, the algorithm outputs an estimation $\hat{x}_v$ of v such that $\mathbb{P}\{|\hat{x}_v - x_v| > \varepsilon(||X||_1 - x_v)\} < \delta$, where $\varepsilon, \delta > 0$ are given as parameters of the algorithm. The estimation is computed by constructing a two-dimensional array C of $t \times k$ counters through a collection of 2-universal hash functions $\{h_\ell\}_{1 \leq \ell \leq t}$, where $k = e/\varepsilon$ and $t = \lceil \log(1/\delta) \rceil$. Each time an item v is read from the input stream, this causes one counter per line to be incremented, *i.e.*, $C[\ell][h_\ell(v)]$ is incremented for all $\ell \in [t]$.

To compute the sketch codeviation of any two streams σ_1 and σ_2, two sketches $\widehat{\sigma}_1$ and $\widehat{\sigma}_2$ of these streams are constructed according to the above description (*i.e.*, construction of two arrays C_{σ_1} and C_{σ_1} of $t \times k$ counters through t 2-universal hash functions $\{h_\ell\}_{1 \leq \ell \leq t}$). Note that there is no particular assumption on the length of both streams σ_1 and σ_2 (their respective length m_1 and m_2 are finite but unknown). By properties of the 2-universal hash functions $\{h_\ell\}_{1 \leq \ell \leq t}$, each line ℓ of C_{σ_1} and C_{σ_2} corresponds to the same partition ρ_ℓ of Ω, and each entry a of line ℓ corresponds to $\hat{X}_{\rho_\ell}(a)$ (*cf.* Definition 1). Therefore, when a query is issued to compute the sketch codeviation $\widehat{\text{cod}}$ between these two streams, the

Algorithm 1: sketch codeviation algorithm

Input: Two input streams σ_1 and σ_2; δ and ε precision settings;

Output: The sketch codeviation $\widehat{\text{cod}}_k(\sigma_1, \sigma_2)$ between σ_1 and σ_2

```
t ← ⌈ln 1/δ⌉; k ← ⌈e/ε⌉;
Choose t functions h : Ω → [k], each from a 2-universal hash function family;
C_σ1[1..t][1..k] ← 0;
C_σ2[1..t][1..k] ← 0;
for i ∈ σ1 do
    for ℓ = 1 to t do
        C_σ1[ℓ][h_ℓ(i)] ← C_σ1[ℓ][h_ℓ(i)] + 1;
for j ∈ σ2 do
    for ℓ = 1 to t do
        C_σ2[ℓ][h_ℓ(j)] ← C_σ2[ℓ][h_ℓ(j)] + 1;
On query cod^(σ1, σ2) return
min_{1≤ℓ≤t} cod(C_σ1[ℓ][−], C_σ2[ℓ][−])
```

codeviation value between the ℓ^{th} line of C_{σ_1} and C_{σ_2} for each $\ell = 1 \ldots t$ is computed, and the minimum value among these t ones is returned. Figure 1 presents the pseudo-code of our algorithm.

Theorem 10 *The sketch codeviation $\widehat{\text{cod}}(X, Y)$ returned by Algorithm 1 satisfies, with $E_{\text{cod}} = \widehat{\text{cod}}(X, Y) - \text{cod}(X, Y)$,*

$$E_{\text{cod}} \geq 0 \textit{ and}$$

$$\mathbb{P}\left\{|E_{\text{cod}}| \geq \frac{\varepsilon}{N}\left(\|X\|_1\|Y\|_1 - \|XY\|_1\right)\right\} \leq \delta.$$

Proof: The first relation holds by Proposition 3. Regarding the second one, let us first consider the ℓ-th line of both C_{σ_1} and C_{σ_2}. We have

$$\begin{aligned}\widehat{\text{cod}}[\ell](X, Y) &= \text{cod}(C_{\sigma_1}[\ell][-], C_{\sigma_2}[\ell][-]) \\ &= \frac{1}{N}\sum_{a=1}^{k} C_{\sigma_1}[\ell][a]C_{\sigma_2}[\ell][a] \\ &\quad - \left(\frac{1}{N}\sum_{a=1}^{k} C_{\sigma_1}[\ell][a]\right)\left(\frac{1}{N}\sum_{a=1}^{k} C_{\sigma_1}[\ell][a]\right).\end{aligned}$$

By construction of Algorithm 1, $\forall 1 \leq \ell \leq t, \forall i, j \in \sigma_1$ such that $h_\ell(i) = h_\ell(j) = a$, we have

$$C_{\sigma_1}[\ell][a] = x_i + \sum_{j \neq i} x_j.$$

Similarly, $\forall 1 \leq \ell \leq t, \forall i, j \in \sigma_2$ such that $h_\ell(i) = h_\ell(j) = a$, we have

$$C_{\sigma_2}[\ell][a] = y_i + \sum_{j \neq i} y_j.$$

Thus,

$$\begin{aligned}\widehat{\text{cod}}[\ell](X, Y) &= \frac{1}{N}\sum_{a=1}^{k}\left(\sum_{\substack{i=1 \\ h_\ell(i)=a}}^{N} x_i\right)\left(\sum_{\substack{i=1 \\ h_\ell(i)=a}}^{N} y_i\right) \\ &\quad - \frac{1}{N}\sum_{a=1}^{k}\left(\sum_{\substack{i=1 \\ h_\ell(i)=a}}^{N} x_i\right)\frac{1}{N}\sum_{a=1}^{k}\left(\sum_{\substack{i=1 \\ h_\ell(i)=a}}^{N} y_i\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} x_i y_i + \frac{1}{N}\sum_{\substack{i \neq j \\ h_\ell(i)=h_\ell(j)}} x_i y_j \\ &\quad - \left(\frac{1}{N}\sum_{i=1}^{N} x_i\right)\left(\frac{1}{N}\sum_{i=1}^{N} y_i\right) \\ &= \text{cod}(X, Y) + \frac{1}{N}\sum_{\substack{i \neq j \\ h_\ell(i)=h_\ell(j)}} x_i y_j\end{aligned}$$

We have

$$\begin{aligned}&\mathbb{E}\left[\widehat{\text{cod}}[\ell](X, Y)\right] \\ &\quad = \mathbb{E}\left[\text{cod}(X, Y)\right] + \frac{1}{N}\sum_{i \neq j} x_i y_j \mathbb{P}\{h_\ell(i) = h_\ell(j)\}\end{aligned}$$

By linearity of the expectation, we get

$$\begin{aligned}&\mathbb{E}\left[\widehat{\text{cod}}[\ell](X, Y) - \text{cod}(X, Y)\right] \\ &\quad = \frac{1}{N}\sum_{i \neq j} x_i y_j \mathbb{P}\{h_\ell(i) = h_\ell(j)\}\end{aligned}$$

By definition of 2-universal hash functions, we have $\mathbb{P}\{h_\ell(i) = h_\ell(j)\} \leq \frac{1}{k}$. Therefore,

$$\begin{aligned}\mathbb{E}\left[\widehat{\text{cod}}[\ell](X, Y) - \text{cod}(X, Y)\right] &\leq \frac{1}{Nk}\sum_{i \neq j} x_i y_j \\ &= \frac{1}{Nk}\left(\|X\|_1\|Y\|_1 - \|XY\|_1\right)\end{aligned}$$

By definition of k (*cf.* Algorithm 1), we have

$$\begin{aligned}&\mathbb{E}\left[\widehat{\text{cod}}[\ell](X, Y) - \text{cod}(X, Y)\right] \\ &\quad \leq \frac{\varepsilon}{eN}\left(\|X\|_1\|Y\|_1 - \|XY\|_1\right)\end{aligned}$$

Using the Markov inequality, we obtain

$$\begin{aligned}&\mathbb{P}\left\{|\widehat{\text{cod}}[\ell](X, Y) - \text{cod}(X, Y)| \geq \frac{\varepsilon}{N}\left(\|X\|_1\|Y\|_1 - \|XY\|_1\right)\right\} \\ &\quad \leq \frac{1}{e}\end{aligned}$$

By construction $\widehat{\text{cod}}(X, Y) = \min_{1 \leq \ell \leq t} \widehat{\text{cod}}[\ell](X, Y)$. Thus, by definition of t (*cf.* Algorithm 1) we obtain

$$\begin{aligned}&\mathbb{P}\left\{|\widehat{\text{cod}}(X, Y) - \text{cod}(X, Y)| \geq \frac{\varepsilon}{N}\left(\|X\|_1\|Y\|_1 - \|XY\|_1\right)\right\} \\ &\quad \leq \left(\frac{1}{e}\right)^t = \delta\end{aligned}$$

that concludes the proof. ■

Lemma 11 *Algorithm 1 uses* $\mathcal{O}\left((\frac{1}{\varepsilon})\log\frac{1}{\delta}(\log N+\log m)\right)$ *bits of space to give an approximation of the sketch codeviation, where* $m=\max(\|X\|_1,\|Y\|_1)$.

Proof: Both matrices C_{σ_i} for $i\in\{1,2\}$ are composed of $t\times k$ counters, where each counter uses $\mathcal{O}(\log m)$ bits of space. With a suitable choice of hash family, we can store each of the t hash functions above in $\mathcal{O}(\log N)$ space. This gives an overall space bound of $\mathcal{O}(t\log N+tk\log m)$, which proves the lemma with the chosen values of k and t. ■

V. Distributed codeviation Approximation Algorithm

In this section, we propose an algorithm that computes the codeviation between a set of n distributed data streams, so that the number of bits communicated between the n sites and the coordinator is minimized. This amounts for the coordinator to compute an approximation of the codeviation matrix Σ, which is the dispersion matrix of the n data streams. Specifically, let $\mathbb{X}=\{X_1,X_2,\dots,X_n\}$ be the set of fingerprint vectors $X_1,\dots,X_n$ describing respectively the streams $\sigma_1,\dots,\sigma_n$. We have

$$\widehat{\Sigma}=\left[\widehat{\mathrm{cod}}(X_i,X_j)\right]_{1\le i\le n,1\le j\le n}.$$

The algorithm proceeds in rounds until all the data streams have been read in their entirety. In the following, we denote by $\sigma_i^{(r)}$ the substream of σ_i received by S_i during the round r, and by d_r the number of data items in this substream.

In a bootstrap phase corresponding to round $r=1$ of the algorithm, each site S_i computes a single sketch C_{σ_i} of the received data stream σ_i as described in lines 5–7 of Algorithm 1. Once node S_i has received d_1 data items (where d_1 should typically be set to 100 [18]), then node S_i sends $C_{\sigma_i^{(1)}}$ to the coordinator, keeps a copy of $C_{\sigma_i^{(1)}}$, and starts a new round $r=2$. Upon receipt of $C_{\sigma_i^{(1)}}$ from any S_i, the coordinator asks all the $n-1$ other nodes S_j to send their own sketch $C_{\sigma_j^{(1)}}$.

Once the coordinator has received all $C_{\sigma_i^{(1)}}$, for $1\le i\le n$, it sets $\forall i\in[n], C_{\sigma_i}\leftarrow C_{\sigma_i^{(1)}}$. The coordinator builds the sketch codeviation matrix $\widehat{\Sigma}=\left[\widehat{\mathrm{cod}}(X_i,X_j)\right]_{1\le i\le n,1\le j\le n}$ such that the element in position i,j is the sketch codeviation between streams σ_i and σ_j. As the codeviation is symmetric, the codeviation matrix is a symmetric matrix, and thus only the upper-triangle and the diagonal need to be computed.

At round $r>1$, each node S_i computes a new sketch $C_{\sigma_i^{(r)}}$ with the sequence of data streams received since the beginning of round r. Let $d_r=2d_{r-1}$ be an upper bound on the number of received items during round r. When node S_i has received at least $d_{r-1}/2$ data items, it starts to compute the sketch codeviation between $C_{\sigma_i^{(r-1)}}$ and $C_{\sigma_i^{(r)}}$ as in line 11 of Algorithm 1. Once node S_i has received d_r data items since the beginning of round r, then it sends its current sketch $C_{\sigma_i^{(r)}}$ to the coordinator and starts a new round $r+1$. Note that during round r, S_i regularly computes $\mathrm{cod}\left(\sigma_i^{(r-1)},\sigma_i^{(r)}\right)$ to detect whether significant variations in the stream have occurred before having received d_r items. This allows to inform the coordinator as quickly as possible that some attack might be undergoing. S_i might then send its current sketch $C_{\sigma_i^{(r)}}$ to the coordinator once $\mathrm{cod}\left(\sigma_i^{(r-1)},\sigma_i^{(r)}\right)$ has reached a sufficiently small value. An interesting question left for future work is the study of such a value. Upon receipt of the first $C_{\sigma_i^{(r)}}$ from any S_i, the coordinator asks all the $n-1$ other nodes S_j to send it their own sketch $C_{\sigma_j^{(r)}}$. The coordinator locally updates the n sketches such as $C_{\sigma_i}\leftarrow C_{\sigma_i}+C_{\sigma_i^{(r)}}$ and updates the codeviation matrix $\widehat{\Sigma}$ on every couple of sketches.

Theorem 12 *The approximated codeviation matrix* $\widehat{\Sigma}$ *returned by the distributed sketch codeviation algorithm satisfies* $\widehat{\Sigma}\ge\Sigma$ *and*

$$\mathbb{P}\left\{\left|\widehat{\Sigma}-\Sigma\right|\ge\frac{\varepsilon}{N}\max_{i,j\in[n]}\left(\|X_i\|_1\|X_j\|_1-\|X_iX_j\|_1\right)\right\}\le\delta.$$

Proof: The statement is derived from Theorem 10 and the fact that the expectation of a matrix is defined as the matrix of expected values. ■

Lemma 13 *The distributed sketch codeviation algorithm gives an approximation of matrix* Σ*, using* $\mathcal{O}\left((1/\varepsilon)\log(1/\delta)\left(\log N+\log m\right)\right)$ *bits of space for each* n *nodes, and* $\mathcal{O}\left(n\log m\left(1/\varepsilon\log(1/\delta)+n\right)\right)$ *bits of space for the coordinator, where* m *is the maximum size among all the streams,* i.e., $m=\max_{i\in[n]}\|X_i\|_1$.

Proof: From the algorithm definition, each node maintains two sketches with space describes in Lemma 11. The coordinator maintains n matrices of $t\times k$ counters and the $n\times n$ codeviation matrix which takes $\mathcal{O}(n^2\log m)$ bits, where $m=\max_{i\in[n]}\|X_i\|_1$. One can note that the coordinator does not need to maintain the t hash functions. ■

Lemma 14 *The distributed sketch codeviation algorithm gives an approximation of matrix* Σ *by sending* $\mathcal{O}\left(rn(1+(1/\varepsilon)\log(m/2)\log(1/\delta))\right)$ *bits, where* r *is the number of the last round and* m *is the maximum size of the streams.*

Proof: Suppose that the number of rounds of the algorithm is equal to r. At each round, the size of the substream on each node is at most doubled, and then lower or equal to $\frac{\|X_i\|_1}{2}$. An upper bound of number of bits sent by any node during a round r is trivially given by $(1/\varepsilon)\log(m/2)\log(1/\delta)$ where $m=\max_{i\in[n]}\|X_i\|_1$. Finally, at each end of round, the coordinator sends 1 bit to at most $n-1$ nodes. ■

VI. Performance Evaluation

We have implemented the distributed sketch codeviation algorithm and have conducted a series of experiments on different types of streams and for different parameters settings. We have fed our algorithm with both real-world data sets and synthetic traces. Real data give a realistic representation of some existing monitoring applications, while the latter ones allow to capture phenomenons which may be difficult to obtain from real-world traces, and thus allow to check the robustness

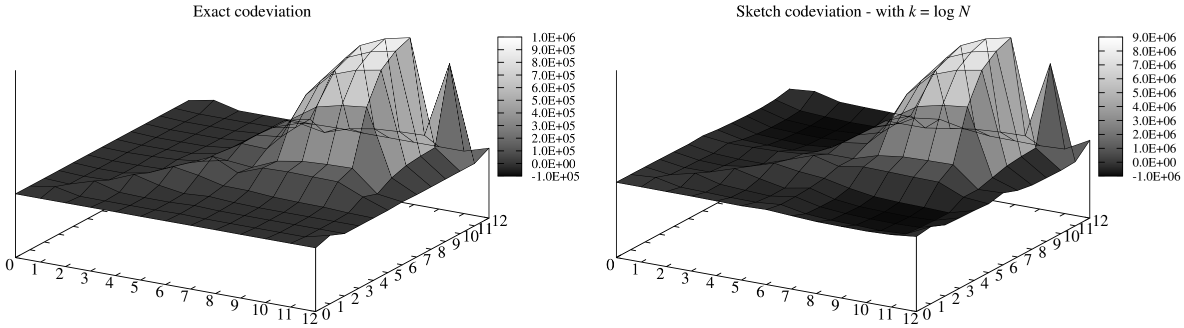


Figure 1. Synthetic traces – The isopleth on the left has been computed with all the items in memory, while the one on the right has been computed by the distributed algorithm from sketches of length $k = \log N$.

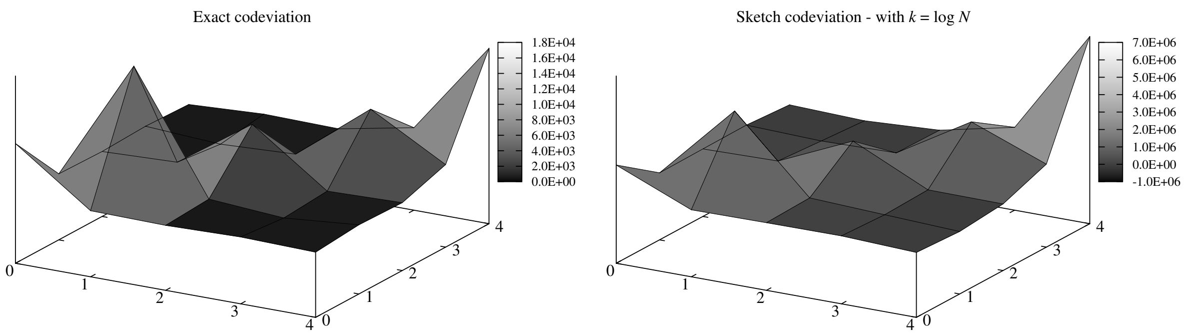


Figure 2. Real datasets – The isopleth on the left has been computed with all the items in memory, while the one on the right has been computed by the distributed algorithm from sketches of length $k = \log N$.

of our metric. Synthetic traces of streams have been generated from 13 distributions showing very different shapes, that is the Uniform distribution (referred to as distribution 0 in the following), the Zipfian or power law one with parameter α from 1 to 5 (referred to as distributions $1, \dots, 5$), the Poisson distribution with parameter λ from $N/2^1$ to $N/2^5$ (distributions $6, \dots, 11$), and the Binomial and the Negative Binomial ones (distributions 12 and 13). All the streams generated from these distributions have a length of around $100,000$ items, and contain no more than $1,000$ distinct items. Real data have been downloaded from the repository of Internet network traffic [25]. We have used 5 large traces among the available ones. Two of them represent two weeks logs of HTTP requests to the Internet service provider ClarkNet WWW server – ClarkNet is a full Internet access provider for the Metro Baltimore-Washington DC area – the other two ones contain two months of HTTP requests to the NASA Kennedy Space Center WWW server, and the last one represents seven months of HTTP requests to the WWW server of the University of

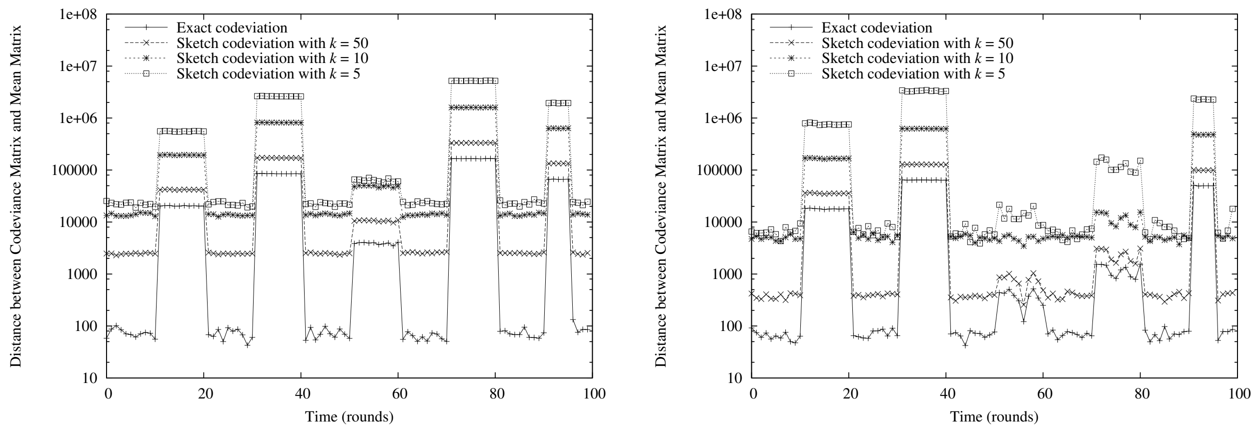
Table I
STATISTICS OF THE FIVE REAL DATA TRACES.

Data trace	Trace	# items (m)	# distinct (n)	max. freq.
NASA (July)	0	1,891,715	81,983	17,572
NASA (August)	1	1,569,898	75,058	6,530
ClarkNet (August)	2	1,654,929	90,516	6,075
ClarkNet (September)	3	1,673,794	94,787	7,239
Saskatchewan	4	2,408,625	162,523	52,695

Saskatchewan, Canada. In the following these data sets will be respectively referred to as ClarkNet, NASA, and Saskatchewan traces. We have used as data items the source hosts of the HTTP requests. Table I presents some statistics of these five data traces, in term of stream size (*cf.* “# items”), number of distinct items in each stream (*cf.* “# distinct”) and the number of occurrences of the most frequent item (*cf.* “max. freq.”).

A. Experimental evaluation of the Sketch codeviation

Figures 1 and 2 summarize the results obtained by feeding our distributed codeviation algorithm with respectively syn-



(a) With $\mathbb{E}(\Sigma_N)$ computed on "normal" traffic behavior

(b) With $\mathbb{E}(\Sigma_r)$ computed on "historical" traffic behavior

Figure 3. Distance between the codeviation matrix and the mean of the past ones when all the 10 synthetic traces follow different distributions as a function of the rounds of the protocol, with $\delta = 10^{-5}$.

thetics traces and real datasets. The isopeths on the left of respectively Figures 1 and 2 represent the $n \times n$ codeviation matrix computed by storing in memory the streams in their entirety. The isopeths on the right of respectively Figures 1 and 2 correspond to the $n \times n$ sketch codeviation matrix returned by the distributed algorithm based on sketches of size $k = \log N$. Both the x-axis and the y-axis represent the 13 synthetic streams on Figure 1, and the 5 real data sets on Figure 2, while the z-axis represents the value of each cell matrix in both figures.

These results clearly show that our distributed algorithm is capable of efficiently and accurately quantifying how observed data streams change together and in which proportion whatever the shape of the input streams. Indeed, by using sketches of size $k = \log N$, one obtains isopeths very similar to the ones computed with all the items stored in memory. Note that the order of magnitude exhibited by the sketch codeviation matrix is due to the overestimation factor and remains proportional to the exact one. Both results from synthetic traces and real datasets lead to the same conclusions. The following experimental results focus on the detection of attacks.

B. Detection of different profiles of attacks

Figure 3 shows how efficiently our approximation distributed algorithm detects different scenarii of attacks in real time. Specifically, we compute at each round of the distributed protocol, the distance between the codeviance matrix Σ constructed from the streams under investigation and the mean of covariance matrices $\mathbb{E}(\Sigma_N)$ computed under normal situations. This distance has been proposed in [6]. Specifically, given two square matrices M and M' of size n, consider the distance as follows:

$$\|M - M'\| = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(M_{i,j} - M'_{i,j})^2}.$$

We evaluate at each round r, the variable d_r defined by

$$d_r = \|\Sigma_r - \mathbb{E}(\Sigma_N)\|.$$

Interestingly, Jin and Yeung [6] propose to detect abnormal behaviors with respect to normal ones as follows. First they analyze normal traffic-wide behaviors, and estimate at the end of analysis, a point c and a constant a for d_r satisfying $\mid d_r - c \mid < a, \forall r \in \mathbb{N}^*$. The constant a is selected as the upper threshold of the i.i.d $\mid d_r - c \mid$. Then when investigating the potential presence of DDoS attacks over the network, they consider as abnormal any traffic pattern that shows for any r, $\mid d_r - c \mid > a$. Because we think that it is not tractable to characterize what is a normal network-wide traffic *a priori*, we adapt this definition by considering the past behavior of the traffic under investigation. Specifically, at any round $r > 1$, the distance is computed between the current codeviance matrix Σ_r and the mean one $\mathbb{E}(\Sigma_r)$ corresponding to previous rounds $1, \ldots, r-1, r$. That is $\mathbb{E}(\Sigma_r) = ((r-1)\mathbb{E}(\Sigma_{r-1}) + \Sigma_r)/r$. As shown in Figure 3(b), this distance provides better results than the ones obtained with the original distance [6], which is depicted in Figure 3(a).

Based on these distances, we have fed our distributed algorithm with different patterns of traffic. Specifically, Figure 3 shows the distance between the codeviance matrix and the mean ones (respectively based on normal ones for Figure 3(a) and on past ones for Figure 3(b)). These distances are depicted, as a function of time, when the codeviance is exactly computed and when it is estimated with our distributed algorithm with different values of k. What can be seen is that, albeit there are up to two orders of magnitude between the exact codeviance matrix and the estimated one, the shape of the codeviance variations are for most of them similar, especially in Figure 3(b). Different attack scenarii are simulated. From round 0 to 10, all the 10 synthetic traces follow the same nominal distribution (*e.g.*, a Poisson distribution). Then from round 10 to 20 a targeted attack is launched by flooding a single node (*i.e.*, one among the ten traces follows a Zipfian distribution with $\alpha = 4$). This gives rise to a drastic and abrupt increase of the distance. As can be shown, the estimated covariance exactly follows the exact one, which is a very good result. Then after coming back to a "normal" traffic, half of the traces are replaced by Zipfian ones (from round 30 to 40), representing a flooding attack toward a group of nodes. As for the previous attack, the covariance matrices are highly impacted by this attack. From round 50 to 60, traces follow a Zipfian distribution with $\alpha = 1$ which represents unbalanced network traffic but should not be completely representative of attacks. On the other hand, in the fourth and fifth attack periods, all the traces follow a Zipfian distribution with different values of $\alpha \geq 2$, which clearly shows a flooding attack toward a group of targeted nodes.

From these experiments, one could extract the value of the upper threshold a. For instance, a should be set to $1,000$ for the exact codeviation and for the sketch codeviation with $k = 50$, which lead to detect all the DDoS attacks. Considering the sketch codeviation with $k = 10$ (respectively $k = 5$), a should be set to $10,000$ (respectively $50,000$) in order to detect all these attacks.

The main lesson drawn from these results is the good performance of our distributed algorithm whatever the pattern of the attack.

VII. Conclusion and Future Works

In this paper we have proposed a novel metric, named the sketch codeviation, that allows to approximate the deviation between any number of distributed streams. We have given upper and lower bounds on the quality of this metric, and have provided an algorithm that additively approximates it using very little space. Beyond its theoretical interest, the sketch codeviation can be exploited in many applications. As discussed in the introducing, large scale monitoring applications are quite straightforward application domains, but we might also use it in publish-subscribe applications, where it must be interesting to track the temporal and spatial correlations that may exist between the different attributes of such applications. This study is planned for future work.

References

[1] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Procs of the ACM Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, 2005.

[2] T. Qiu, Z. Ge, D. Pei, J. Wang, and J. Xu, "What happened in my network: mining network events from router syslogs," in *Procs of the 10th ACM conference on Internet measurement (IMC)*, 2010.

[3] D. S. Yeung, "Covariance-matrix modeling and detecting various flooding attacks," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 37, no. 2, pp. 157–169, 2007.

[4] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On flow correlation attacks and countermeasures in mix networks," in *Procs of the 4th ACM International Conference on Privacy Enhancing Technologies (PET)*, 2004.

[5] S. Ganguly, M. Garafalakis, R. Rastogi, and K. Sabnani, "Streaming algorithms for robust, real-time detection of ddos attacks," in *Procs of the 27th International Conference on Distributed Computing Systems (ICDCS '07)*, 2007.

[6] S. Jin and D. Yeung, "A covariance analysis model for ddos attack detection," in *4th IEEE International Conference on Communications (ICC '04)*, vol. 4, 2004, pp. 1882–1886.

[7] E. Anceaume, Y. Busnel, and S. Gambs, "Uniform and Ergodic Sampling in Unstructured Peer-to-Peer Systems with Malicious Nodes," in *Procs of the 14th International Conference on Principles of Distributed Systems (OPODIS)*, 2010.

[8] Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan, "Counting distinct elements in a data stream," in *6th International Workshop on Randomization and Approximation Techniques (RANDOM)*. Springer-Verlag, 2002, pp. 1–10.

[9] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," *Journal of Computer and System Sciences*, vol. 31, no. 2, pp. 182–209, 1985.

[10] D. M. Kane, J. Nelson, and D. P. Woodruff, "An optimal algorithm for the distinct element problem," in *Symposium on Principles of Databases (PODS)*, 2010.

[11] N. Alon, Y. Matias, and M. Szegedy, "The space complexity of approximating the frequency moments," in *Procs of the 22th ACM Symposium on Theory of computing (STOC)*, 1996.

[12] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," *Theoretical Computer Science*, vol. 312, no. 1, pp. 3–15, 2004.

[13] A. Chakrabarti, G. Cormode, and A. McGregor, "A near-optimal algorithm for computing the entropy of a stream," in *Procs of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[14] A. Lall, V. Sekar, M. Ogihara, J. Xu, and H. Zhang, "Data streaming algorithms for estimating entropy of network traffic," in *Procs of the Joint International ACM Conference on Measurement and Modeling of Computer systems (SIGMETRICS)*, 2006.

[15] E. Anceaume, Y. Busnel, and S. Gambs, "On the power of the adversary to solve the node sampling problem," *Transactions on Large-Scale Data- and Knowledge-Centered Systems (TLDKS)*, vol. 11, pp. 102–126, 2013.

[16] E. Anceaume and Y. Busnel, "A distributed information divergence estimation over data streams," *IEEE Transactions on Parallel Distributed Systems (TPDS)*, vol. 25, no. 2, pp. 478–487, 2014.

[17] G. Cormode, S. Muthukrishnan, and K. Yi, "Algorithms for distributed functional monitoring," in *Procs of the 19th Annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, 2008.

[18] C. Arackaparambil, J. Brody, and A. Chakrabarti, "Functional monitoring without monotonicity," in *Procs of the 36th ACM International Colloquium on Automata, Languages and Programming (ICALP)*, 2009.

[19] P. B. Gibbons and S. Tirthapura, "Estimating simple functions on the union of data streams," in *13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2001, pp. 281–291.

[20] Z. Haung, K. Yi, and Q. Zhang, "Randomized algorithms for tracking distributed count, frequencies and ranks," in *Proc. of 31st ACM Symposium on Principles of Database Systems (PODS)*, 2012.

[21] Z. Liu, B. Radunovic, and M. Vojnovic, "Continuous distributed counting for non-monotonic streams," in *Proc. of 31st ACM Symposium on Principles of Database Systems (PODS)*, 2012.

[22] J. Yuan and K. Mills, "Monitoring the macroscopic effect of DDoS flooding attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 4, 2005.

[23] Muthukrishnan, *Data Streams: Algorithms and Applications*. Now Publishers Inc., 2005.

[24] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *Journal of Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[25] the Internet Traffic Archive, "http://ita.ee.lbl.gov/html/traces.html," Apr. 2008.