Lightweight Metric Computation for Distributed Massive Data Streams

Emmanuelle Anceaume, Yann Busnel

Lightweight Metric Computation for Distributed Massive Data Streams

Emmanuelle Anceaume[1] and Yann Busnel[2⋆]

[1] IRISA / CNRS Rennes (France), emmanuelle.anceaume@irisa.fr
[2] IMT Atlantique, Rennes (France), yann.busnel@imt-atlantique.fr

Abstract. The real time analysis of massive data streams is of utmost importance in data intensive applications that need to detect as fast as possible and as efficiently as possible (in terms of computation and memory space) any correlation between its inputs or any deviance from some expected nominal behavior. The IoT infrastructure can be used for monitoring any events or changes in structural conditions that can compromise safety and increase risk. It is thus a recurrent and crucial issue to determine whether huge data streams, received at monitored devices, are correlated or not as it may reveal the presence of attacks. We propose a metric, called codeviation, that allows to evaluate the correlation between distributed massive streams. This metric is inspired from classical metric in statistics and probability theory, and as such enables 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 then generalize our approach by presenting a new metric, the *Sketch-⋆ metric*, which allows us to define a distance between updatable summaries of large data streams. An important feature of the *Sketch-⋆ metric* is that, given a measure on the entire initial data streams, the *Sketch-⋆ metric* preserves the axioms of the latter measure on the sketch. We finally present results obtained during extensive experiments conducted on both synthetic traces and real data sets allowing us to validate the robustness and accuracy of our metrics.

Keywords: Data stream model; correlation metric; statistical metric; distributed approximation algorithm

1 Introduction

Performance of many complex monitoring applications, including Internet monitoring applications, and data mining, or massively distributed infrastructures

⋆ This work has been partially funded by the French ANR project SocioPlug (ANR-13-INFR-0003) and by the DeSceNt project granted by the Labex CominLabs excellence laboratory (ANR-10-LABX-07-01)

such as sensor networks, and the Internet of Things (IoT) 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 intrusion detection tools [3,4,5], while analyzing the effect of multivariate correlation should help for an early detection of Distributed Denial of Service (DDoS) [6]. Finally, the sustainable development of smart cities is expected to handle large amounts of data generated from large number of sensors with the consequent necessity for quick aggregation of the data, which could be exploited to detect correlated events. Among possible applications, smart building management systems rely on service-oriented continuous queries over sensor data streams in case of energy consumption monitoring [7], air pollution monitoring applications heavily rely on sensors to detect threshold crossings [8]. More generally, Stankovic [9] argues that the real time analysis of large and distributed data streams is of utmost importance to tackle issues related to creative knowledge, robustness, privacy, and security.

The point is that, in all these contexts, 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. Actually, 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 for an exact computation of 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 [10]. In contrast, the streaming approach consists in scanning, on the fly, each piece of data of the input stream, and in locally keeping only compact synopses or sketches that contain the most important information about these data. This approach enables the derivation of some data streams statistics with guaranteed error bounds without making any assumptions on the order in which data items are received at nodes.

Work on data stream analysis mainly focuses on efficient methods (data-structures and algorithms) to answer different kind of queries over massive data streams, as for example the computation of the number of different data items in a given stream [11,12,13]. Mostly, these methods consist in deriving statistic estimators over the data stream, in creating summary representations of streams (to build histograms, wavelets, and quantiles), and in comparing data streams. Regarding the construction of estimators, a seminal work is due to Alon *et al.* [14]. The authors have proposed estimators of the frequency moments F_k of

a stream, which are important statistical tools that allow to quantify specificities of a data stream. Subsequently, a lot of attention has been devoted to the strongly related notion of the entropy [15] of a stream [16,17,18], and all notions based on entropy as the quantification of the amount of randomness of a stream (*e.g*, [17,19,20,21]). The construction of synopses or sketches of the data stream have been proposed for different applications (*e.g*, [22,23,24,25]). Actually in [26], the authors propose a characterization of the information divergences that are not sketchable, and prove that any distance that has not "norm-like" properties is not sketchable.

On the other hand, very few works have tackled the distributed streaming model, also called the functional monitoring problem [27], 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 [27,28,29]. Cormode *et al.* [27] 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 [14]. Arackaparambil *et al.* [28] consider the empirical entropy estimation [14] 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 [30,31].

We go a step further by studying the dispersion matrix of distributed streams. Specifically, we propose a novel metric that allows us 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 [32], such a network-wide traffic monitoring tool should allow monitoring applications to get significant information on the traffic behavior changes to subsequently inform more detailed detection tools on where DDoS attacks are currently active. We provide a distributed algorithm that additively approximates the codeviation among n data streams $\sigma_1, \ldots, \sigma_n$ by using a sublinear number of bits of space for each of the n nodes, sublinear in the domain size from which items values are drawn, and in the largest size of these data streams.

We then generalize our approach by proposing a novel metric, named *Sketch-$\star$ metric* in the following, that reflects the relationships between any two massive data streams. Actually, the problem of detecting changes or outliers in a data stream is similar to the problem of identifying patterns that do not conform to an expected behavior, which has been an active area of research for many decades. To accurately analyze streams of data, a panel of information-theoretic measures and distances have been proposed as key measures in statistical inference and data processing problems [33]. There exist two broad classes of measures, namely

the f-divergences, introduced by Csiszar, Morimoto and Ali & Silvey [34,35,36], and the Bregman divergences, which are very important to quantify the amount of information that separates two distributions. Among them, the most commonly used are the Kullback-Leibler (KL) divergence [37], the Jensen-Shannon divergence and the Battacharyya distance [38]. More details can be found in the comprehensive survey of Basseville [33].

Unfortunately, computing information theoretic measures of distances in the data stream model is challenging essentially because one needs to process a huge amount of data sequentially, on the fly, and by using very little storage with respect to the size of the stream. In addition the analysis must be robust over time to detect any sudden change in the observed streams.

We tackle this issue with the *Sketch-⋆ metric*. This metric allows us to efficiently and accurately estimate a broad class of distance measures between any two large data streams. Such an estimation is achieved by computing these distances only on compact synopses or sketches of streams. The *Sketch-⋆ metric* is distribution-free and makes no assumption about the underlying data volume. It is thus capable of comparing any two data streams, identifying their correlation if any, and more generally, it allows us to acquire a deep understanding of the structure of the input streams. Formalization of this metric is one of the contributions of this paper. We present an approximation algorithm that constructs a sketch of the stream from which the *Sketch-⋆ metric* is computed. As for the codeviation, this algorithm is a one-pass algorithm. It uses very basic computations, little storage space (*i.e.*, logarithmic in the size of the input stream and the number of items in the stream), and does not need any information on the structure of the input stream.

Road Map of the Paper. In Section 2, we present the computational model under which we analyze our algorithms and derive bounds, and recall some mathematical background that will be needed in the remaining of the paper.

We present in Section 3 the sketch codeviation that allows us to approximate in real time the correlation between distributed and massive streams. We give upper and lower bounds on the quality of this approximated metric with respect to the codeviation in Section 3.2. 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 in Section 3.3 the algorithm that computes the sketch codeviation between any two data streams. We extend this algorithm to handle distributed streams. Section 3.4 presents our distributed algorithm 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 $\varepsilon m/N$, as shown by performance evaluation results presented in Section 3.5.

The *Sketch-⋆ metric*, which allows us to efficiently estimate a broad class of distances measures between any two large data streams by computing these distances only using compact synopses or sketches of the streams is introduced in Section 4. Formalization of the *Sketch-⋆ metric* is presented in Section 4.2. The description of the algorithm that approximates the *Sketch-⋆ metric* in one pass appears in Section 4.3. This algorithm uses very basic computations, little storage space (*i.e.*, $\mathcal{O}(t(\log N + k \log m))$ where k and t are precision parameters, and m and N are respectively the size of the input stream and the domain size from which items values are drawn), and does not need any information on the structure of the input stream. Finally, the robustness of our approach is validated with a detailed experimentation study based on both synthetic traces that range from stable streams to highly skewed ones, and real data sets.

2 Data Stream Model

2.1 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 [27]. We assume that communication is instantaneous. We refer the reader to [39] for a detailed description of data streaming models and algorithms. Note that in the IoT context, it may not be reasonnable to rely on a central entity. We could extend our distributed solution to a fully decentralized version by organizing sites in such a way that each one could locally aggregate the information provided by its neighbours, as done in [40].

2.2 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 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 in the following by abusing

the notation, we denote this "$|\Omega|$-point distribution" by "Ω-point distribution", also known as the item frequency vector of σ. Note also 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 $m = \|X\|_1$. A natural approach to study a data stream σ is to model it as an empirical data distribution over the universe Ω, given by $(p_1, p_2, \ldots, p_N)$ with $p_i = x_i/m$.

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 $x \in [M]$.

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 $\mathbb{P}\{|\hat{\phi} - \phi| > \varepsilon\} < \delta$, where $\varepsilon, \delta > 0$ are given as parameters of the algorithm.

3 Correlation estimation using Codeviation

3.1 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 3.4. We propose a metric over Ω-point distributions 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 3.5, this metric captures shifts in the network-wide traffic behavior when a DDoS attack is active. The codeviation between any two Ω-point distributions $X = (x_1, x_2, \ldots, x_N)$, and $Y = (y_1, y_2, \ldots, y_N)$ is the real number denoted $\mathrm{cod}(X, Y)$ defined by

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

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

3.2 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 Ω-point distributions 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*

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

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 1. *Let $X = (x_1, \ldots, x_N)$, and $Y = (y_1, \ldots, y_N)$ be any two Ω-point distributions. 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 1. *The sketch codeviation is a function of the codeviation. We have*

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

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

Proof. From Relation (1), we have

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

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.

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 1 (Accuracy of the sketch codeviation). *Let X and Y be any two Ω-point distributions of items, such that $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_N)$. If $k \geq |\operatorname{supp}(X) \cap \operatorname{supp}(Y)| + \mathbf{1}_{\operatorname{supp}(X) \setminus \operatorname{supp}(Y)} + \mathbf{1}_{\operatorname{supp}(Y) \setminus \operatorname{supp}(X)}$ then*

$$\widehat{\operatorname{cod}}_k(X,Y) = \operatorname{cod}(X,Y),$$

where $\operatorname{supp}(X)$*, respectively* $\operatorname{supp}(Y)$*, 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 $k = |\operatorname{supp}(X) \cap \operatorname{supp}(Y)| + \mathbf{1}_{\operatorname{supp}(X) \setminus \operatorname{supp}(Y)} + \mathbf{1}_{\operatorname{supp}(Y) \setminus \operatorname{supp}(X)}$. We consider a partition $\overline{\rho} \in \mathcal{P}_k(\Omega)$ defined as follows

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

  Then from Relation (2) we have

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

  Thus, $\sum_{a \in \overline{\rho}} \sum_{i \in a} \sum_{j \in a \setminus \{i\}} x_i y_j = 0$. From Proposition (1), we get that $\widehat{\operatorname{cod}}_k(X,Y) = \operatorname{cod}(X,Y)$.
- **Case 2:**
  For $k > |\operatorname{supp}(X) \cap \operatorname{supp}(Y)| + \mathbf{1}_{\operatorname{supp}(X) \setminus \operatorname{supp}(Y)} + \mathbf{1}_{\operatorname{supp}(Y) \setminus \operatorname{supp}(X)}$ (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. □

Derivation of Upper Bounds on $\mathcal{E}_k(X,Y)$ We have shown with Theorem 1 that the sketch codeviation matches exactly the codeviation if $k \geq |\operatorname{supp}(X) \cap \operatorname{supp}(Y)| + \mathbf{1}_{\operatorname{supp}(X) \setminus \operatorname{supp}(Y)} + \mathbf{1}_{\operatorname{supp}(Y) \setminus \operatorname{supp}(X)}$. 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 Ω-point distribution 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 $\mathcal{X} = \{X = (x_1, \ldots, x_N) \text{ such that } ||X||_1 = m_{\mathcal{X}}\}$ and $\mathcal{Y} = \{Y = (y_1, \ldots, y_N) \text{ such that } ||Y||_1 = m_{\mathcal{Y}}\}$. The following theorem derives the maximum value of the overestimation factor.

Theorem 2 (Upper bound of $\mathcal{E}_k(X,Y)$). *Let $k \geq 1$ be the precision parameter of the sketch codeviation. For any two Ω-point distributions $X \in \mathcal{X}$ 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. For readability reason, the proof of this theorem is presented in Appendix A. □

Theorem 2 shows 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 now demonstrate that, given X and Y, the overestimation factor $\mathcal{E}_k(X,Y)$ is a decreasing function in k.

Lemma 2. *Let X and Y be any two Ω-point distributions. 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

$$\mathcal{E}_1(X,Y) = \frac{1}{N} \sum_{i \in \Omega} \sum_{j \in \Omega \setminus \{i\}} x_i y_j. \tag{3}$$

– **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

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

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 1 concludes the proof. □

3.3 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 [41]. 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 $\hat{x}_v$

Algorithm 1: Sketch codeviation algorithm

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

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

$t \leftarrow \lceil \ln \frac{1}{\delta} \rceil$; $k \leftarrow \lceil \frac{e}{\varepsilon} \rceil$;
Choose t functions $h : \Omega \to [k]$, each from a 2-universal hash function family;
$C_{\sigma_1}[1..t][1..k] \leftarrow 0$;
$C_{\sigma_2}[1..t][1..k] \leftarrow 0$;
for $i \in \sigma_1$ **do**
  for $\ell = 1$ **to** t **do**
    $C_{\sigma_1}[\ell][h_\ell(i)] \leftarrow C_{\sigma_1}[\ell][h_\ell(i)] + 1$;
for $j \in \sigma_2$ **do**
  for $\ell = 1$ **to** t **do**
    $C_{\sigma_2}[\ell][h_\ell(j)] \leftarrow C_{\sigma_2}[\ell][h_\ell(j)] + 1$;
On query $\widehat{\mathrm{cod}}(\sigma_1, \sigma_2)$ **return** $\min_{1 \le \ell \le t} \mathrm{cod}(C_{\sigma_1}[\ell][-], C_{\sigma_2}[\ell][-])$

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 $\{h_\ell\}_{1 \le \ell \le t}$, 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 $\{h_\ell\}_{1 \le \ell \le t}$). 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 $\{h_\ell\}_{1 \le \ell \le t}$, 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 $\widehat{\mathrm{cod}}$ between these two streams, the codeviation value between the ℓ^{th} line of C_{σ_1} and C_{σ_2} for each $\ell = 1 \dots t$ is computed, and the minimum value among these t ones is returned. Figure 1 presents the pseudo-code of our algorithm.

Theorem 3. *The sketch codeviation* $\widehat{\mathrm{cod}}(X, Y)$ *returned by Algorithm 1 satisfies, with* $E_{\mathrm{cod}} = \widehat{\mathrm{cod}}(X, Y) - \mathrm{cod}(X, Y)$,

$$E_{\mathrm{cod}} \ge 0 \text{ and } \mathbb{P}\left\{|E_{\mathrm{cod}}| \ge \frac{\varepsilon}{N}\left(\|X\|_1\|Y\|_1 - \|XY\|_1\right)\right\} \le \delta.$$

Proof. The first relation holds by Proposition 1. Regarding the second one, let us first consider the ℓ-th line of both C_{σ_1} and C_{σ_2}. We have

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

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,

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

We have

$$\mathbb{E}\left[\widehat{\mathrm{cod}}[\ell](X,Y)\right] = \mathbb{E}\left[\mathrm{cod}(X,Y)\right] + \frac{1}{N}\sum_{i\neq j} x_i y_j \mathbb{P}\{h_\ell(i) = h_\ell(j)\}.$$

By linearity of the expectation, we get

$$\mathbb{E}\left[\widehat{\mathrm{cod}}[\ell](X,Y) - \mathrm{cod}(X,Y)\right] = \frac{1}{N}\sum_{i\neq j} x_i y_j \mathbb{P}\{h_\ell(i) = h_\ell(j)\}.$$

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

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

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

$$\mathbb{E}\left[\widehat{\mathrm{cod}}[\ell](X,Y) - \mathrm{cod}(X,Y)\right] \leq \frac{\varepsilon}{eN}\left(\|X\|_1\|Y\|_1 - \|XY\|_1\right)$$

Using the Markov inequality, we obtain

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

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

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

that concludes the proof. □

Lemma 3. *Algorithm 1 uses* $\mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)\log\frac{1}{\delta}(\log N + \log m)\right)$ *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. □

3.4 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. As previously evoked in Section 2, it is possible to have a fully decentralized version of our algorithm, by for example, organizing the sites along a distributed hash table (DHT) and by taking profit of the additive property of the Count-Min data structure to allow each site to aggregate their schetch so have to progressively obtain a global view of the system. Such a possible solution appears in [40]. In the following we present the coordinator-based version for clarity of the analysis. Note however that the distributed version would have a non negligeable impact on the communication cost. This issue is left for future work.

Specifically, let $\mathbb{X} = \{X_1, X_2, \ldots, X_n\}$ be the set of Ω-point distributions $X_1, \ldots, X_n$ describing respectively the streams $\sigma_1, \ldots, \sigma_n$. We have

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

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 [28]), 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 $1 \leq i \leq n$, it sets $\forall i \in [n], C_{\sigma_i} \leftarrow C_{\sigma_i^{(1)}}$. The coordinator builds the sketch codeviation matrix $\widehat{\Sigma} = \left[\widehat{\mathrm{cod}}(X_i, X_j)\right]_{1 \leq i \leq n, 1 \leq j \leq n}$ 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 4. *The approximated codeviation matrix $\widehat{\Sigma}$ returned by the distributed sketch codeviation algorithm satisfies $\widehat{\Sigma} \geq \Sigma$ and*

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

Proof. The statement is derived from Theorem 3 and the fact that the expectation of a matrix is defined as the matrix of expected values. □

Lemma 4 (Space complexity). *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 3. 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 5 (Communication complexity). *The distributed sketch codeviation algorithm gives an approximation of matrix Σ using a communication complexity of $\mathcal{O}\left(rn(1 + (1/\varepsilon)\log(m/2)\log(1/\delta))\right)$ 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. □

Lemma 6 (Time complexity). *The time complexity of sketch codeviation is $\mathcal{O}(\log 1/\delta)$ per update in the reading phase of the stream, and $\mathcal{O}\left(1/\varepsilon \log 1/\delta\right)$ per query.*

Proof. Based on the pseudo-code provided in Algorithm 1, an update requires to hash the item, then retrieve and increase a cell for each row, thus the update time complexity is $O(\log 1/\delta)$. On the other hand, a query requires to sum the scalar product of each row, by retrieving each cell of the both data structure. The query time complexity is then $O(1/\varepsilon \log 1/\delta)$. □

3.5 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

Table 1. 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

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 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 $1, \ldots, 5$), the Poisson distribution with parameter λ from $N/2^1$ to $N/2^5$ (distributions $6, \ldots, 11$), 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 [42]. 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 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 1 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."). Note that all these benchmarks share a Zipfian behavior, with a lower α for the University of Saskatchewan.

Experimental evaluation of the Sketch codeviation Figures 1 and 2 summarize the results obtained by feeding our distributed codeviation algorithm with respectively synthetics 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

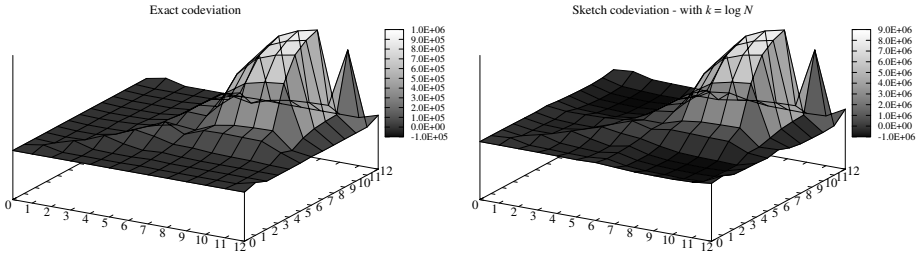


Fig. 1. Synthetic traces – The isopleth on the left has been computed with all the items in memory, while the one on the right has been computed by the distributed algorithm from sketches of length $k = \log N$.

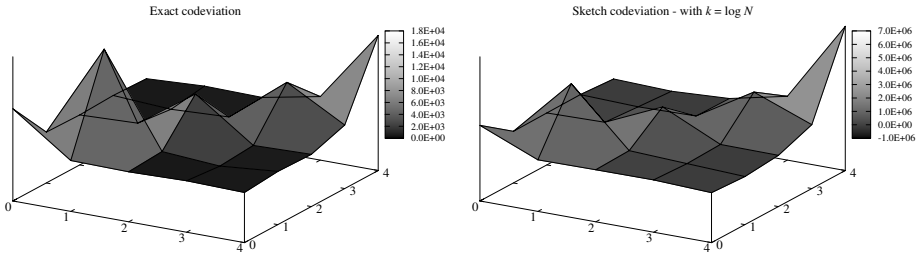
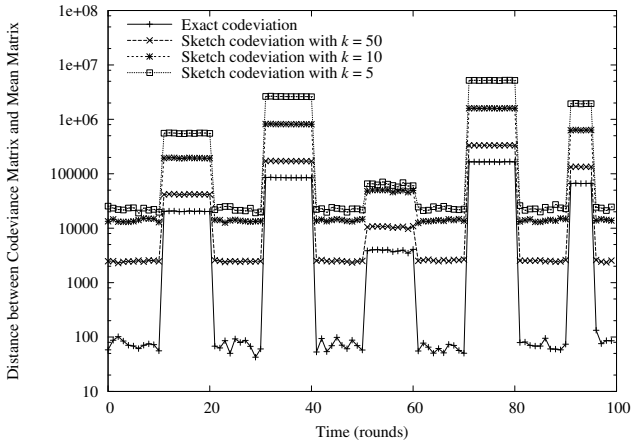


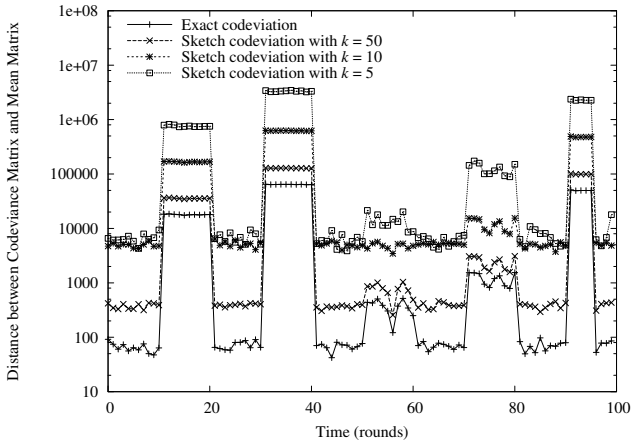
Fig. 2. Real datasets – The isopleth on the left has been computed with all the items in memory, while the one on the right has been computed by the distributed algorithm from sketches of length $k = \log N$.

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.



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



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

Fig. 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 $\delta = 10^{-5}$.

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 $|d_r - c| < a, \forall r \in \mathbb{N}^*$. The constant a is selected as the upper threshold of the i.i.d $|d_r - c|$. Then when investigating the potential presence of DDoS attacks over the network, they consider as abnormal any traffic pattern that shows for any r, $|d_r - c| > a$. 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.

4 Sketch-$\star$ metric

We generalize the above approach by proposing the Sketch-$\star$ metric that reflects the relationships between any two discrete probability distributions in the context of massive data streams. To accurately analyze streams of data, a panel of information-theoretic measures and distances have been proposed to answer the specificities of the analyses. Among them, the most commonly used are the Kullback-Leibler (KL) divergence [37], or more generically, the f-divergences, introduced by Csiszar, Morimoto and Ali & Silvey [34,35,36], the Jensen-Shannon divergence and the Battacharyya distance [38]. After having recalled the formal definitions of these metrics, we introduce the Sketch-$\star$ metric specification, and then present a space and computation-efficient algorithm to compute any generalized metric ϕ between the summaries of any two stream σ_1 and σ_2, such that this computation preserves all the properties of ϕ computed on σ_1 and σ_2. We finally show the robustness of our approach through extensive simulations.

4.1 Metrics and divergences

This section is devoted to the description of a collection of metrics.

Metric definitions The classical definition of a metric is based on a set of four axioms.

Definition 2 (Metric). *Given a set X, a metric is a function $d\colon X \times X \to \mathbb{R}$ such that, for any $x, y, z \in X$, we have:*

$$\textit{Non-negativity:} \quad d(x,y) \geq 0 \tag{4}$$

$$\textit{Identity of indiscernibles:} \quad d(x,y) = 0 \Leftrightarrow x = y \tag{5}$$

$$\textit{Symmetry:} \quad d(x,y) = d(y,x) \tag{6}$$

$$\textit{Triangle inequality:} \quad d(x,y) \leq d(x,z) + d(z,y) \tag{7}$$

In the context of information divergence, usual distance functions are not precisely metric. Indeed, most of divergence functions do not verify the 4 axioms, but only a subset of them. For instance, a pseudometric is a function that verifies the axioms of a metric with the exception of the identity of indiscernible, while a premetric is a pseudometric that relax both the symmetry and the triangle inequality axioms.

Definition 3 (Pseudometric). *Given a set X, a* ***pseudometric*** *is a function that verifies the axioms of a metric with the exception of the* identity of indiscernible*, which is replaced by*

$$\forall x \in X, d(x, x) = 0.$$

Note that this definition allows that $d(x, y) = 0$ for some $x \neq y$ in X.

Definition 4 (Quasimetric). *Given a set X, a* ***quasimetric*** *is a function that verifies all the axioms of a metric with the exception of the* symmetry *(*cf. *Relation 6).*

Definition 5 (Semimetric). *Given a set X, a* ***semimetric*** *is a function that verifies all the axioms of a metric with the exception of the* triangle inequality *(*cf. *Relation 7).*

Definition 6 (Premetric). *Given a set X, a* ***premetric*** *is a pseudometric that relax both the* symmetry *and* triangle inequality *axioms.*

Definition 7 (Pseudoquasimetric). *Given a set X, a* ***pseudoquasimetric*** *is a function that relax both the* identity of indiscernible *and the* symmetry *axioms.*

Note that the latter definition simply corresponds to a premetric satisfying the triangle inequality. Remark also that all the generalized metrics preserve the *non-negativity* axiom.

Two classes of generalized metrics, usually denoted as *divergences*, that allow to measure the separation of distributions have been proposed, namely the class of f-divergences and the class of Bregman divergences.

***f*-divergence** The class of f-divergences provides a set of relations that is used to measure the "distance" between two distributions p and q. Mostly used in the context of statistics and probability theory, a f-divergence $\mathcal{D}_f$ is a premetric that guarantees monotonicity and convexity.

Definition 8 (f-divergence). *Let p and q be two Ω-point distributions. Given a convex function $f : (0, \infty) \to \mathbb{R}$ such that $f(1) = 0$, the f-divergence of q from p is*

$$\mathcal{D}_f(p||q) = \sum_{i \in \Omega} q_i f\left(\frac{p_i}{q_i}\right),$$

where by convention, we assume that $0f(\frac{0}{0}) = 0$, $af(\frac{0}{a}) = a \lim_{u \to 0} f(u)$, and $0f(\frac{a}{0}) = a \lim_{u \to \infty} f(u)/u$ if these limits exist.

Property 1 (Monotonicity). Given κ an arbitrary transition probability that respectively transforms two Ω-point distributions p and q into p_κ and q_κ, we have:

$$\mathcal{D}_f(p||q) \geq \mathcal{D}_f(p_\kappa||q_\kappa).$$

Property 2 (Convexity). Let p_1, p_2, q_1 and q_2 be four Ω-point distributions. Given any $\lambda \in [0,1]$, we have:

$$\mathcal{D}_f\left(\lambda p_1 + (1-\lambda)p_2||\lambda q_1 + (1-\lambda)q_2\right) \leq \lambda \mathcal{D}_f(p_1||q_1) + (1-\lambda)\mathcal{D}_f(p_2||q_2).$$

Bregman divergence Initially proposed in [43], the Bregman divergences are a generalization of the notion of distance between points. This class of generalized metrics always satisfies the non-negativity and identity of indecernibles. However they do not always satisfy the triangle inequality and their symmetry depends on the choice of the differentiable convex function F. Specifically,

Definition 9 (Bregman divergence (BD)). *Given a continuously-differentiable and strictly convex function F defined on a closed convex set C, the Bregman divergence of p from q is*

$$\mathcal{B}_F(p||q) = F(p) - F(q) - \langle \nabla F(q), (p-q) \rangle .$$

where the operator $\langle \cdot, \cdot \rangle$ denotes the inner product, and $\nabla F(q)$ is the gradient of F at q.

In the context of data stream, it is possible to reformulate this definition as follows. Specifically,

Definition 10 (Decomposable BD).
Let p and q be any two Ω-point distributions. Given a strictly convex function $F : (0,1] \to \mathbb{R}$, the Bregman divergence of q from p is defined as

$$\mathcal{B}_F(p||q) = \sum_{i \in \Omega} \left(F(p_i) - F(q_i) - (p_i - q_i)F'(q_i)\right).$$

The Bregman divergence verifies non-negativity and convexity properties in its first argument, but not necessarily in the second argument. Another interesting property is given by thinking of the Bregman divergence as an operator of the function F.

Property 3 (Linearity). Let F_1 and F_2 be any two strictly convex and differentiable functions. Given any $\lambda \in [0,1]$, we have that

$$\mathcal{B}_{F_1+\lambda F_2}(p||q) = \mathcal{B}_{F_1}(p||q) + \lambda \mathcal{B}_{F_2}(p||q).$$

Classical metrics Based on these definitions, we present several commonly used metrics in Ω-point distribution context. These specific metrics are used in the evaluation part presented in Section 4.4.

Kullback-Leibler divergence The Kullback-Leibler (KL) divergence [37], also called the relative entropy, is a robust metric for measuring the statistical difference between two data streams. The KL divergence owns the special feature that it is both a f-divergence and a Bregman one (with $f(t) = F(t) = t \log t$).

Given p and q two Ω-point distributions, the Kullback-Leibler divergence is defined as

$$\mathcal{D}_{KL}(p||q) = \sum_{i \in \Omega} p_i \log \frac{p_i}{q_i}. \tag{8}$$

Jensen-Shannon divergence The Jensen-Shannon divergence (JS) is a symmetrized version of the Kullback-Leibler divergence. Also known as information radius (IRad) or total divergence to the average, it is defined as

$$\mathcal{D}_{JS}(p||q) = \frac{1}{2} D_{KL}(p||\ell) + \frac{1}{2} D_{KL}(q||\ell), \tag{9}$$

where $\ell = \frac{1}{2}(p + q)$. Note that the square root of this divergence is a metric.

Bhattacharyya distance The Bhattacharyya distance is derived from his proposed measure of similarity between two multinomial distributions, also known as the Bhattacharya coefficient (BC) [38]. It is a semimetric as it does not verify the triangle inequality. It is defined as

$$\mathcal{D}_B(p||q) = -\log(BC(p,q)) \text{ where } BC(p,q) = \sum_{i \in \Omega} \sqrt{p_i q_i}.$$

Note that the famous Hellinger distance [44] is equal to $\sqrt{1 - BC(p,q)}$ verifies it.

4.2 Sketch-★ metric

We now present a method to sketch two input data streams σ_1 and σ_2, and to compute any generalized metric ϕ between these sketches such that this computation preserves all the properties of ϕ computed on σ_1 and σ_2.

Definition 11 (Sketch-$\star$ metric). *Let p and q be any two Ω-point distributions. Given a precision parameter k, and any generalized metric ϕ on the set of all Ω-point distributions, there exists a Sketch-$\star$ metric $\widehat{\phi}_k$ defined as follows*

$$\widehat{\phi}_k(p||q) = \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{q}_\rho).$$

We recall that, again, $\forall a \in \rho$, $\widehat{p}_\rho(a) = \sum_{i \in a} p_i$ and where $\mathcal{P}_k(\Omega)$ is the set of all partitions of Ω into exactly k nonempty and mutually exclusive cells.

Remark 1. Note that for $k > N$, it does not exist a partition of Ω into k nonempty parts. By convention, we consider that $\widehat{\phi}_k(p||q) = \phi(p||q)$ in this specific context.

In this section, we focus on the preservation of axioms and properties of a generalized metric ϕ by the corresponding Sketch-$\star$ metric $\widehat{\phi}_k$.

Axioms preserving

Theorem 5. *Given any generalized metric ϕ then, for any $k \in \mathbb{N}$, the corresponding Sketch-$\star$ metric $\widehat{\phi}_k$ preserves all the axioms of ϕ.*

Proof. The proof is directly derived from Lemmata 7, 8, 9 and 10. □

Lemma 7 (Non-negativity). *Given any generalized metric ϕ verifying the Non-negativity axiom then, for any $k \in \mathbb{N}$, the corresponding* Sketch-$\star$ metric $\widehat{\phi}_k$ *preserves the Non-negativity axiom.*

Proof. Let p and q be any two Ω-point distributions. By definition,

$$\widehat{\phi}_k(p||q) = \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{q}_\rho)$$

As for any two k-point distributions, ϕ is positive we have $\widehat{\phi}_k(p||q) \geq 0$ that concludes the proof. □

Lemma 8 (Identity of indiscernible). *Given any generalized metric ϕ verifying the Identity of indiscernible axiom then, for any $k \in \mathbb{N}$, the corresponding* Sketch-$\star$ metric $\widehat{\phi}_k$ *preserves the Identity of indiscernible axiom.*

Proof. Let p be any Ω-point distribution. We have

$$\widehat{\phi}_k(p||p) = \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{p}_\rho) = 0,$$

due to the Identity of indiscernible axiom on ϕ.

Consider now two Ω-point distributions p and q such that $\widehat{\phi}_k(p||q) = 0$. Metric ϕ verifies both the non-negativity axiom (by construction) and the Identity of indiscernible axiom (by assumption). Thus we have $\forall \rho \in \mathcal{P}_k(\Omega), \widehat{p}_\rho = \widehat{q}_\rho$, leading to

$$\forall \rho \in \mathcal{P}_k(\Omega), \forall a \in \rho, \sum_{i \in a} p(i) = \sum_{i \in a} q(i). \tag{10}$$

Moreover, for any $i \in \Omega$, there exists a partition $\rho \in \mathcal{P}_k(\Omega)$ such that $\{i\} \in \rho$. By Equation 10, $\forall i \in \Omega, p(i) = q(i)$, and so $p = q$.

Combining the two parts of the proof leads to $\widehat{\phi}_k(p||q) = 0 \iff p = q$, which concludes the proof of the Lemma. □

Lemma 9 (Symmetry). *Given any generalized metric ϕ verifying the Symmetry axiom then, for any $k \in \mathbb{N}$, the corresponding* Sketch-$\star$ metric $\widehat{\phi}_k$ *preserves the Symmetry axiom.*

Proof. Let p and q be any two Ω-point distributions. We have

$$\widehat{\phi}_k(p||q) = \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{q}_\rho).$$

Let $\overline{\rho} \in \mathcal{P}_k(\Omega)$ be a k-cell partition such that $\phi(\widehat{p}_{\overline{\rho}}||\widehat{q}_{\overline{\rho}}) = \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{q}_\rho)$. We get

$$\widehat{\phi}_k(p||q) = \phi(\widehat{p}_{\overline{\rho}}||\widehat{q}_{\overline{\rho}}) = \phi(\widehat{q}_{\overline{\rho}}||\widehat{p}_{\overline{\rho}}) \leq \widehat{\phi}_k(q||p).$$

By symmetry, considering $\underline{\rho} \in \mathcal{P}_k(\Omega)$ such that $\phi(\widehat{q}_{\underline{\rho}}||\widehat{p}_{\underline{\rho}}) = \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{q}_\rho||\widehat{p}_\rho)$, we also have $\widehat{\phi}_k(q||p) \leq \widehat{\phi}_k(p||q)$, which concludes the proof. □

Lemma 10 (Triangle inequality). *Given any generalized metric ϕ verifying the Triangle inequality axiom then, for any $k \in \mathbb{N}$, the corresponding* Sketch-$\star$ metric $\widehat{\phi}_k$ *preserves the Triangle inequality axiom.*

Proof. Let p, q and r be any three Ω-point distributions. Let $\overline{\rho} \in \mathcal{P}_k(\Omega)$ be a k-cell partition such that $\phi(\widehat{p}_{\overline{\rho}}||\widehat{q}_{\overline{\rho}}) = \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{q}_\rho)$. We have

$$\begin{aligned}
\widehat{\phi}_k(p||q) &= \phi(\widehat{p}_{\overline{\rho}}||\widehat{q}_{\overline{\rho}}) \\
&\leq \phi(\widehat{p}_{\overline{\rho}}||\widehat{r}_{\overline{\rho}}) + \phi(\widehat{r}_{\overline{\rho}}||\widehat{q}_{\overline{\rho}}) \\
&\leq \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{r}_\rho) + \max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{r}_\rho||\widehat{q}_\rho) \\
&= \widehat{\phi}_k(p||r) + \widehat{\phi}_k(r||q)
\end{aligned}$$

that concludes the proof. □

Properties preserving

Theorem 6. *Given a f-divergence ϕ then, for any $k \in \mathbb{N}$, the corresponding Sketch-$\star$ metric $\widehat{\phi}_k$ is also a f-divergence.*

Proof. From Theorem 5, $\widehat{\phi}_k$ preserves the axioms of the generalized metric. Thus, $\widehat{\phi}_k$ and ϕ are in the same equivalence class. Moreover, from Lemma 11, $\widehat{\phi}_k$ verifies the monotonicity property. Thus, as the f-divergence is the only class of decomposable information *monotonic* divergences (*cf.* [35]), $\widehat{\phi}_k$ is also a f-divergence. □

Theorem 7. *Given a Bregman divergence ϕ then, for any $k \in \mathbb{N}$, the corresponding Sketch-$\star$ metric $\widehat{\phi}_k$ is also a Bregman divergence.*

Proof. From Theorem 5, $\widehat{\phi}_k$ preserves the axioms of the generalized metric. Thus, $\widehat{\phi}_k$ and ϕ are in the same equivalence class. Moreover, the Bregman divergence is characterized by the property of transitivity (*cf.* [45]) defined as follows. Given p, q and r three Ω-point distributions such that $q = \Pi(L|r)$ and $p \in L$, with Π is a selection rule according to the definition of Csiszár in [45] and L is a subset of the Ω-point distributions, we have the Generalized Pythagorean Theorem:

$$\phi(p||q) + \phi(q||r) = \phi(p||r).$$

Moreover the authors in [46] show that the set Γ_N of all discrete probability distributions over N elements ($\{x_1, \ldots, x_N\}$) is a Riemannian manifold, and it

owns another different dually flat affine structure. They also show that these dual structures give rise to the generalized Pythagorean theorem. This is verified for the coordinates in Γ_N and for the dual coordinates [46]. Combining these results with the projection theorem [45,46], we obtain that

$$\begin{aligned}\widehat{\phi}_k(p||r) &= \max_{\rho\in\mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{r}_\rho)\\ &= \max_{\rho\in\mathcal{P}_k(\Omega)} (\phi(\widehat{p}_\rho||\widehat{q}_\rho) + \phi(\widehat{q}_\rho||\widehat{r}_\rho))\\ &= \max_{\rho\in\mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{q}_\rho) + \max_{\rho\in\mathcal{P}_k(\Omega)} \phi(\widehat{q}_\rho||\widehat{r}_\rho)\\ &= \widehat{\phi}_k(p||q) + \widehat{\phi}_k(q||r)\end{aligned}$$

Finally, by the characterization of Bregman divergence through transitivity [45], and reinforced with Lemma 13 statement, $\widehat{\phi}_k$ is also a Bregman divergence. □

In the following, we show that the Sketch-$\star$ metric preserves the properties of divergences.

Lemma 11 (Monotonicity). *Given any generalized metric ϕ verifying the Monotonicity property then, for any $k \in \mathbb{N}$, the corresponding Sketch-$\star$ metric $\widehat{\phi}_k$ preserves the Monotonicity property.*

Proof. Let p and q be any two Ω-point distributions. Given $c < N$, consider a partition $\mu \in \mathcal{P}_c(\Omega)$. As ϕ is monotonic, we have $\phi(p||q) \geq \phi(\widehat{p}_\mu||\widehat{q}_\mu)$ [47]. We split the proof into two cases:

Case (1). Suppose that $c \geq k$. Computing $\widehat{\phi}_k(\widehat{p}_\mu||\widehat{q}_\mu)$ amounts in considering only the k-cell partitions $\rho \in \mathcal{P}_k(\Omega)$ that verify

$$\forall b \in \mu, \exists a \in \rho : b \subseteq a.$$

These partitions form a subset of $\mathcal{P}_k(\Omega)$. The maximal value of $\phi(\widehat{p}_\rho||\widehat{q}_\rho)$ over this subset cannot be greater than the maximal value over the whole $\mathcal{P}_k(\Omega)$. Thus we have

$$\widehat{\phi}_k(p||q) = \max_{\rho\in\mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{q}_\rho) \geq \widehat{\phi}_k(\widehat{p}_\mu||\widehat{q}_\mu).$$

Case (2). Suppose now that $c < k$. By definition, we have $\widehat{\phi}_k(\widehat{p}_\mu||\widehat{q}_\mu) = \phi(\widehat{p}_\mu||\widehat{q}_\mu)$. Consider $\rho' \in \mathcal{P}_k(\Omega)$ such that $\forall a \in \rho', \exists b \in \mu, a \subseteq b$. It then exists a transition probability that respectively transforms $\widehat{p}_{\rho'}$ and $\widehat{q}_{\rho'}$ into $\widehat{p}_\mu$ and $\widehat{q}_\mu$. As ϕ is monotonic, we have

$$\begin{aligned}\widehat{\phi}_k(p||q) &= \max_{\rho\in\mathcal{P}_k(\Omega)} \phi(\widehat{p}_\rho||\widehat{q}_\rho)\\ &\geq \phi(\widehat{p}_{\rho'}||\widehat{q}_{\rho'})\\ &\geq \phi(\widehat{p}_\mu||\widehat{q}_\mu) = \widehat{\phi}_k(\widehat{p}_\mu||\widehat{q}_\mu).\end{aligned}$$

Finally for any value of c, $\widehat{\phi}_k$ guarantees the monotonicity property. This concludes the proof. □

Lemma 12 (Convexity). *Given any generalized metric ϕ verifying the Convexity property then, for any $k \in \mathbb{N}$, the corresponding Sketch-$\star$ metric $\widehat{\phi}_k$ preserves the Convexity property.*

Proof. Let p_1, p_2, q_1 and q_2 be any four Ω-point distributions. Given any $\lambda \in [0,1]$, we have:

$$\begin{aligned}&\widehat{\phi}_k\left(\lambda p_1 + (1-\lambda)p_2 || \lambda q_1 + (1-\lambda)q_2\right)\\ &= \max_{\rho \in \mathcal{P}_k(\Omega)} \phi\left(\lambda \widehat{p_1}_\rho + (1-\lambda)\widehat{p_2}_\rho || \lambda \widehat{q_1}_\rho + (1-\lambda)\widehat{q_2}_\rho\right)\end{aligned}$$

Let $\overline{\rho} \in \mathcal{P}_k(\Omega)$ such that

$$\begin{aligned}&\phi\left(\lambda \widehat{p_1}_{\overline{\rho}} + (1-\lambda)\widehat{p_2}_{\overline{\rho}} || \lambda \widehat{q_1}_{\overline{\rho}} + (1-\lambda)\widehat{q_2}_{\overline{\rho}}\right)\\ &= \max_{\rho \in \mathcal{P}_k(\Omega)} \phi\left(\lambda \widehat{p_1}_\rho + (1-\lambda)\widehat{p_2}_\rho || \lambda \widehat{q_1}_\rho + (1-\lambda)\widehat{q_2}_\rho\right).\end{aligned}$$

As ϕ verifies the Convexity property, we have:

$$\begin{aligned}&\widehat{\phi}_k\left(\lambda p_1 + (1-\lambda)p_2 || \lambda q_1 + (1-\lambda)q_2\right)\\ &= \phi\left(\lambda \widehat{p_1}_{\overline{\rho}} + (1-\lambda)\widehat{p_2}_{\overline{\rho}} || \lambda \widehat{q_1}_{\overline{\rho}} + (1-\lambda)\widehat{q_2}_{\overline{\rho}}\right)\\ &\leq \lambda\phi(\widehat{p_1}_{\overline{\rho}} || \widehat{q_1}_{\overline{\rho}}) + (1-\lambda)\phi(\widehat{p_2}_{\overline{\rho}} || \widehat{q_2}_{\overline{\rho}})\\ &\leq \lambda\left(\max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p_1}_\rho || \widehat{q_1}_\rho)\right) + (1-\lambda)\left(\max_{\rho \in \mathcal{P}_k(\Omega)} \phi(\widehat{p_2}_\rho || \widehat{q_2}_\rho)\right)\\ &= \lambda\widehat{\phi}_k(p_1||q_1) + (1-\lambda)\widehat{\phi}_k(p_2||q_2)\end{aligned}$$

that concludes the proof. □

Lemma 13 (Linearity). *The Sketch-$\star$ metric definition preserves the Linearity property.*

Proof. Let F_1 and F_2 be two strictly convex and differentiable functions, and any $\lambda \in [0,1]$. Consider the three Bregman divergences generated respectively from F_1, F_2 and $F_1 + \lambda F_2$.

Let p and q be two Ω-point distributions. We have:

$$\begin{aligned}\widehat{\mathcal{B}}_{F_1+\lambda F_2{}_k}(p||q) &= \max_{\rho \in \mathcal{P}_k(\Omega)} \mathcal{B}_{F_1+\lambda F_2}(\widehat{p}_\rho || \widehat{q}_\rho)\\ &= \max_{\rho \in \mathcal{P}_k(n)} \left(\mathcal{B}_{F_1}(\widehat{p}_\rho || \widehat{q}_\rho) + \lambda\mathcal{B}_{F_2}(\widehat{p}_\rho || \widehat{q}_\rho)\right)\\ &\leq \widehat{\mathcal{B}}_{F_1{}_k}(p||q) + \lambda\widehat{\mathcal{B}}_{F_2{}_k}(p||q)\end{aligned}$$

As F_1 and F_2 are two strictly convex functions, and taken a leaf out of the Jensen's inequality, we have:

$$\begin{aligned}\widehat{\mathcal{B}}_{F_1{}_k}(p||q) + \lambda\widehat{\mathcal{B}}_{F_2{}_k}(p||q) &\leq \max_{\rho \in \mathcal{P}_k(\Omega)} \left(\mathcal{B}_{F_1}(\widehat{p}_\rho || \widehat{q}_\rho) + \lambda\mathcal{B}_{F_2}(\widehat{p}_\rho || \widehat{q}_\rho)\right)\\ &= \widehat{\mathcal{B}}_{F_1+\lambda F_2{}_k}(p||q)\end{aligned}$$

Algorithm 2: *Sketch-⋆ metric* algorithm

Input: Two input streams σ_1 and σ_2; the distance ϕ, k and t settings;
Output: The distance $\hat{\phi}$ between σ_1 and σ_2

Choose t functions $h : \Omega \to [k]$, each from a 2-universal hash function family;
$C_{\sigma_1}[1...t][1...k] \leftarrow 0$;
$C_{\sigma_2}[1...t][1...k] \leftarrow 0$;
for $i \in \sigma_1$ **do**
  for $\ell = 1$ **to** t **do**
    $C_{\sigma_1}[\ell][h_\ell(i)] \leftarrow C_{\sigma_1}[\ell][h_\ell(i)] + 1$;
for $j \in \sigma_2$ **do**
  for $\ell = 1$ **to** t **do**
    $C_{\sigma_2}[\ell][h_\ell(j)] \leftarrow C_{\sigma_2}[\ell][h_\ell(j)] + 1$;
On query $\hat{\phi}_k(\sigma_1||\sigma_2)$ **return** $\max_{1\leq\ell\leq t}\phi(\mathrm{C}_{\sigma_1}[\ell][-], \mathrm{C}_{\sigma_2}[\ell][-])$;

that concludes the proof. □

To summarize, we have shown that the *Sketch-⋆ metric* preserves all the axioms of a metric as well as the properties of f-divergences and Bregman divergences. We now show how to efficiently implement such a metric.

4.3 Approximation algorithm

In this section, we propose an algorithm that computes the *Sketch-⋆ metric* in one pass on the stream.

To compute the *Sketch-⋆ metric* of two streams σ_1 and σ_2, two sketches $\widehat{\sigma}_1$ and $\widehat{\sigma}_2$ of these streams are constructed as in Section 3.3. Note that again there is no particular assumption on the length of both streams σ_1 and σ_2. That is their respective length is finite but unknown. Figure 2 presents the pseudo-code of our algorithm.

Lemma 14. *Given parameters k and t, Algorithm 2 gives an approximation of the Sketch-⋆ metric, using*

$$\mathcal{O}\left(t(\log N + k \log m)\right) \text{ bits of space.}$$

Proof. The matrices C_{σ_i}, for any $i \in \{1, 2\}$, are composed of $t \times k$ counters, which uses $\mathcal{O}(\log m)$. On the other hand, with a suitable choice of hash family, we can store the hash functions above in $\mathcal{O}(t \log N)$ space. □

4.4 Performance Evaluation

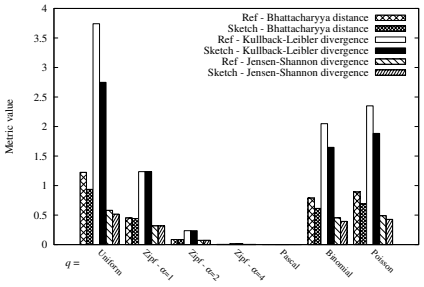
Settings of the experiments We have also implemented our *Sketch-⋆ metric* and have conducted a series of experiments on different types of streams and for different parameters settings. We have fed our algorithm with both real-world

data sets and synthetic traces. We have varied all the significant parameters of our algorithm, that is, the maximal number of distinct data items N in each stream, the number of cells k of each generated partition, and the number of generated partitions t. For each parameters setting, we have conducted and averaged 100 trials of the same experiment, leading to a total of more than $300,000$ experiments for the evaluation of our metric. As in Section 3.5, we feed our algorithm with the same synthetic traces and the real data downloaded from the repository of Internet network traffic [42].

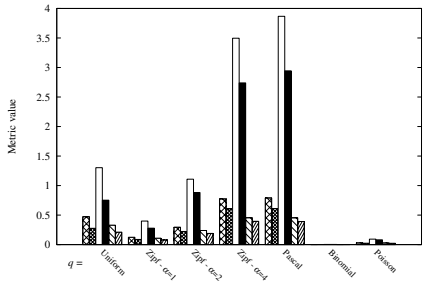
Main lessons drawn from the experiments In this section, we evaluate the accuracy of the *Sketch-⋆ metric* by comparing $\widehat{\phi}_k(p||q)$ with $\phi_k(p||q)$, for $\phi \in \{\text{Kullback-Leiber, Jensen-Shannon, Bhattacharyya}\}$, and for p and q generated from the 7 distributions and the 5 real data sets. Distances computed from the sketches of the stream are referred to as *Sketch* in the legend of the graphs, while the ones computed from the full streams are mentioned as *Ref.* Due to space constraints, only a subset of the results are presented in the paper.

Figure 4 shows the accuracy of our metric as a function of the different input streams and the different generalized metrics applied on these streams. The first noticeable remark is that *Sketch-⋆ metric* behaves perfectly well when the two compared streams follow the same distribution, whatever the generalized metric ϕ used. This can be observed from both synthetic traces (*cf.* Figure 4(a) with both p and q following the Pascal distribution, Figure 4(b) with both p and q following the Binomial distribution, Figure 4(c) with both p and q following the Zipf–$\alpha = 1$ distribution, and Figure 4(d) with both p and q uniformly distributed), and real data sets (*cf.* Figures 4(e) and 4(f) with the NASA (July and August) and ClarkNet (August and September) traces).

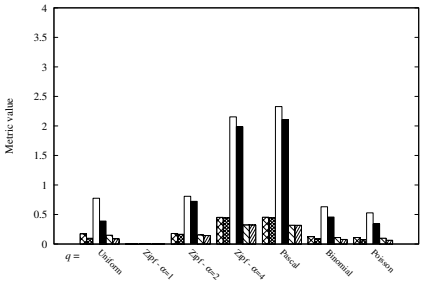
This tendency is further observed when the distributions of input streams are close to each other (*e.g.*, Zipf–$\alpha = 2, 4$ and Pascal distributions, or Uniform and Zipf–$\alpha = 1$). This makes the *Sketch-⋆ metric* a very good candidate as a parametric method for making distribution parameters inference. Another interesting result is shown when the two input streams exhibit a totally different shape. Specifically, let us consider Figures 4(a) and 4(d). Sketching the Uniform distribution leads to k-cell partitions whose value is well distributed, that is, for a given partition ϕ, all the k cell values have with high probability the same value. Now, when sketching the Pascal distribution, the repartition of the data items in the cells of any given partitions is such that a few number of data items (those with high frequency) populate a very few number of cells. However, the values of these cells is very large compared to the other cells, which are populated by a large number of data items whose frequency is small. Thus, the contribution of data items exhibiting a small frequency and sharing the cells of highly frequent items is biased compared to the contribution of the other items. Thus although the input streams show a totally different shape, the accuracy of $\widehat{\phi}_k$ is only slightly lowered in these scenarios which makes it a very powerful tool to compare any two different data streams. The same observation holds with real data sets. When the shapes of the input streams are different (which is the



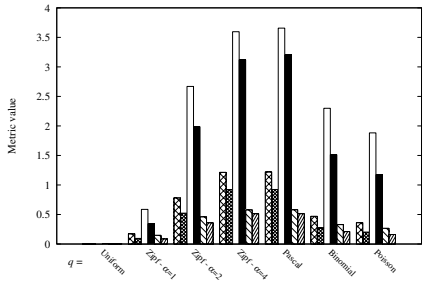
(a) Synthetic traces – Distribution p follows a Negative Binomial NB(3; 0.99) (or Pascal) distribution



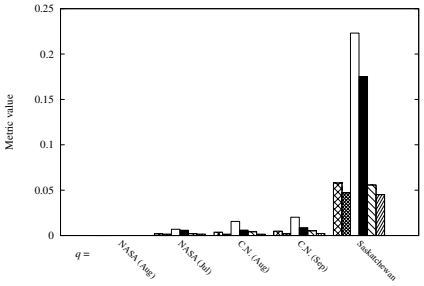
(b) Synthetic traces – Distribution p follows a Binomial distribution with parameter equals to 0.5



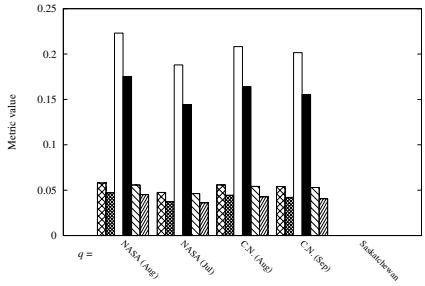
(c) Synthetic traces – Distribution p follows a Zipf distribution with $\alpha = 1$



(d) Synthetic traces – Distribution p follows a Uniform distribution



(e) Real datasets – The input stream p is the NASA (August) trace



(f) Real datasets – The input stream p is the Saskatchewan trace

Fig. 4. Comparison between the *Sketch-⋆ metric* and the ϕ metric as a function of the input stream q either generated from a distribution or real traces. For synthetic traces, $m = 200,000$ and $N = 4,000$. Parameters of the count-min sketch data structure are $k = 200$ and $t = 4$. All the histograms share the same legend, but for readability reasons, this legend is only indicated on histogram 4(a).

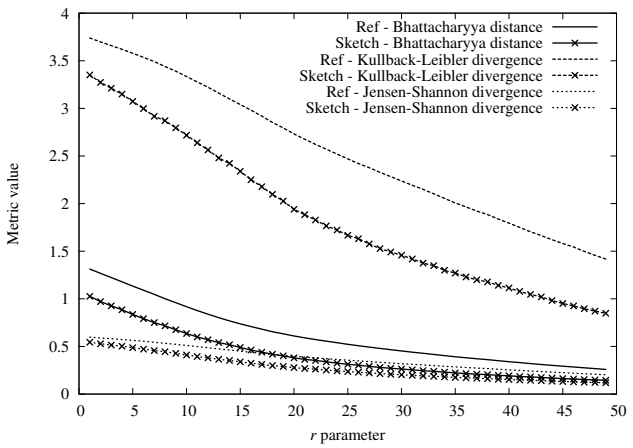


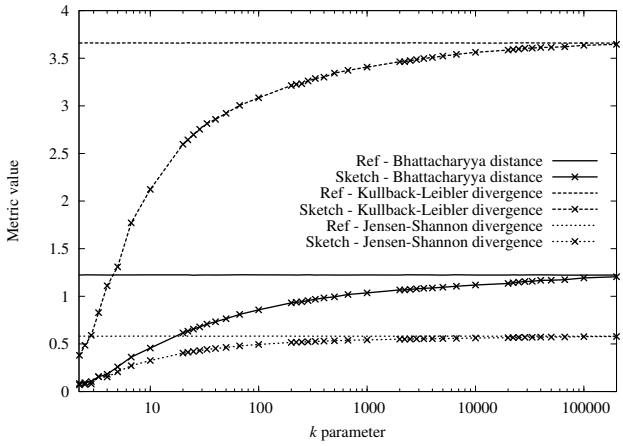
Fig. 5. Comparison between the *Sketch-⋆ metric* and the ϕ metric as a function of the parameters of the Negative Binomial distribution $\mathrm{NB}(r, N/(2r+N))$, where distribution p follows a Uniform distribution and q follows the Negative Binomial distribution $\mathrm{NB}(r, N/(2r+N))$.

case for Saskatchewan with respect to the 4 other input streams), the accuracy of the *Sketch-⋆ metric* decreases a little bit but in a very small proportion. Notice that the scales on the y-axis differ significantly in Figures 4(a)–4(d) and in Figures 4(e)–4(f).

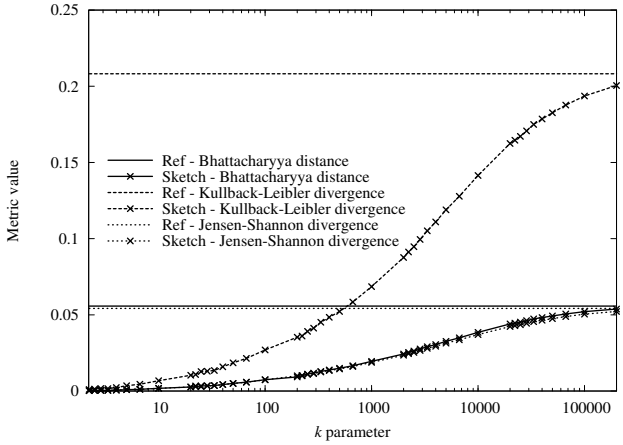
We can also observe the strong impact of the non-symmetry of the Kullback-Leibler divergence on the computation of the distance (computed on full streams or on sketches) with a clear influence when the input streams follow a Pascal and Zipf–$\alpha = 1$ distributions (see Figures 4(a) and 4(c)).

Figure 5 summarizes the good properties of $\widehat{\phi}_k$ by illustrating how, for any generalized metric ϕ, and for any variations in the shape of the two input distributions, $\widehat{\phi}_k$ remains close to ϕ. Recall that increasing values of the r parameter of the Negative Binomial distribution makes the shape of the distribution flatter, while maintaining the same mean value.

Figure 6 presents the impact of the number of cells per generated partition on the accuracy of the-⋆ metric on both synthetic traces and real data. It clearly shows that by increasing k the number of data items per cell in the generated partition shrinks and thus the absolute error on the computation of the distance decreases. The same feature appears when the number N of distinct data items in the stream increases. Indeed, when N increases (for a given k), the number data items per cell augments and thus the precision of our metric decreases. This gives rise to a shift of the inflection point, as illustrated in Figure 6(b) as data sets have almost twenty to forty times more distinct data items than the synthetic ones. As aforementioned, the input streams exhibit very different shapes which explain the strong impact of k. Note also that k has the same influence on the *Sketch-⋆ metric* for all the generalized distances ϕ.



(a) Synthetic traces – Distribution p follows a Uniform distribution and q follows a Negative Binomial NB$(3; 0.99)$ one



(b) Real datasets – The input stream p is the ClarkNet (August) trace and q is the Saskatchewan one

Fig. 6. Comparison between the *Sketch-$\star$ metric* and the ϕ metric as a function of the number of cells k per partition (the number of partitions t of the count-min sketch data structure is set to 4). For synthetic traces, $m = 200,000$ and $N = 4,000$.

Finally, it is interesting to note that the number t of generated partitions has a slight influence on the accuracy of our metric The reason comes from the use of 2-universal hash functions, which guarantee for each of them and with high probability that data items are uniformly distributed over the cells of any partition. As a consequence, augmenting the number of such hash functions has a weak influence on the accuracy of the metric.

5 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 introduction, large scale monitoring applications are quite straightforward application domains, but we might also use it in Internet of Things applications, where it must be interesting to track the temporal and spatial correlations that may exist between the different streams produced by devices in such applications.

In order to generalized this approach, we have introduced another new metric, the *Sketch-$\star$ metric*, that allows to compute any generalized metric ϕ on the summaries of two large input streams. We have presented a simple and efficient algorithm to sketch streams in the same way and compute this metric on these sketches. We have then shown that it behaves pretty well whatever the considered input streams. We are convinced of the indisputable interest of such a metric in various domains including Internet of Things statistical usages as network monitoring and information retrieval [7], and we think that it should pertinent in machine learning, and data mining applications as discussed in [9].

Regarding future works, we plan to characterize our metric among Rényi divergences [48], also known as α-divergences, which generalize different divergence classes. We also plan to consider a fully distributed setting, where each site would be in charge of analyzing its own streams and then would propagate its results to the other sites of the system for comparison or merging (without any coordinator). An immediate application of such a tool would be to detect massive attacks in a decentralized manner (*e.g.*, by identifying specific connection profiles as with worms propagation, and massive port scan attacks or by detecting sudden variations in the volume of received data), which perfectly fits with IoT constraints.

References

1. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: Proceedings of the ACM Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM). (2005)

2. Qiu, T., Ge, Z., Pei, D., Wang, J., Xu, J.: What happened in my network: mining network events from router syslogs. In: Procs of the 10th ACM conference on Internet measurement (IMC). (2010)
3. Yeung, D.S.: Covariance-matrix modeling and detecting various flooding attacks. IEEE Transactions on Systems, Man and Cybernetics, Part A **37**(2) (2007) 157–169
4. Zhu, Y., Fu, X., Graham, B., Bettati, R., Zhao, W.: 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, Garafalakis, M., Rastogi, R., Sabnani, K.: Streaming algorithms for robust, real-time detection of ddos attacks. In: Procs of the 27th International Conference on Distributed Computing Systems (ICDCS). (2007)
6. Jin, S., Yeung, D.: A covariance analysis model for ddos attack detection. In: 4th IEEE International Conference on Communications (ICC). Volume 4. (2004) 1882–1886
7. Pinarer, O., Gripay, Y., Servigne, S., Ozgovde, A.: Energy Enhancement of Multi-application Monitoring Systems for Smart Buildings. In: Conference on Advanced Information Systems Engineering - EnBIS: Energy-awareness and Big Data Management in Information Systems (CAiSE). Volume 249., Ljubljana, Slovenia, Springer (June 2016) 131–142
8. Boubrima, A., Matigot, F., Bechkit, W., Rivano, H., Ruas, A.: Optimal Deployment of Wireless Sensor Networks for Air Pollution Monitoring. In: 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, United States (August 2015)
9. Stankovic, J.A.: Research directions for the internet of things. IEEE Internet of Things Journal **1**(1) (Feb 2014) 3–9
10. Anceaume, E., Busnel, Y., Gambs, S.: Uniform and Ergodic Sampling in Unstructured Peer-to-Peer Systems with Malicious Nodes. In: Proceedings of the 14th international conference on Principles of distributed systems (OPODIS). Volume 6490. (2010) 64–78
11. Bar-Yossef, Z., Jayram, T.S., Kumar, R., Sivakumar, D., Trevisan, L.: Counting distinct elements in a data stream. In: Proceedings of the 6th International Workshop on Randomization and Approximation Techniques (RANDOM), Springer-Verlag (2002) 1–10
12. Flajolet, P., Martin, G.N.: Probabilistic counting algorithms for data base applications. Journal of Computer and System Sciences **31**(2) (1985) 182–209
13. Kane, D.M., Nelson, J., Woodruff, D.P.: An optimal algorithm for the distinct element problem. In: Proceedings of the Symposium on Principles of Databases (PODS). (2010)
14. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (STOC). (1996) 20–29
15. Cover, T., Thomas, J.: Elements of information theory. Wiley New York (1991)
16. Chakrabarti, A., Cormode, G., McGregor, A.: A near-optimal algorithm for computing the entropy of a stream. In: In ACM-SIAM Symposium on Discrete Algorithms. (2007) 328–335
17. Lall, A., Sekar, V., Ogihara, M., Xu, J., Zhang, H.: Data streaming algorithms for estimating entropy of network traffic. In: Proceedings of the joint international conference on Measurement and modeling of computer systems (SIGMETRICS), ACM (2006)

18. Anceaume, E., Busnel, Y., Gambs, S.: On the power of the adversary to solve the node sampling problem. Transactions on Large-Scale Data- and Knowledge-Centered Systems (TLDKS) **11** (2013) 102–126
19. Anceaume, E., Busnel, Y.: An information divergence estimation over data streams. In: Proceedings of the 11th IEEE International Symposium on Network Computing and Applications (NCA). (2012)
20. Chakrabarti, A., Ba, K.D., Muthukrishnan, S.: Estimating entropy and entropy norm on data streams. In: In Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science (STACS), Springer (2006)
21. Guha, S., McGregor, A., Venkatasubramanian, S.: Streaming and sublinear approximation of entropy and information distances. In: Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). (2006) 733–742
22. Rivetti, N., Busnel, Y., Querzoni, L.: Load-aware shedding in stream processing systems. In: Proceedings of the 10th ACM International Conference on Distributed Event-Based Systems (DEBS), Ivine, CA, USA (June 2016)
23. Rivetti, N., Anceaume, E., Busnel, Y., Querzoni, L., Sericola, B.: Online scheduling for shuffle grouping in distributed stream processing systems. In: Proceedings of the 17th ACM/IFIP/USENIX 13th International Conference on Middleware (Middleware), Trento, Italie (December 2016)
24. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. Theoretical Computer Science **312**(1) (2004) 3–15
25. Cormode, G., Garofalakis, M.: Sketching probabilistic data streams. In: Proceedings of the 2007 ACM SIGMOD international conference on Management of data. (2007) 281–292
26. Guha, S., Indyk, P., Mcgregor, A.: Sketching information divergences. Machine Learning **72**(1-2) (2008) 5–19
27. Cormode, G., Muthukrishnan, S., Yi, K.: Algorithms for distributed functional monitoring. In: Procs of the 19th Annual ACM-SIAM Symposium On Discrete Algorithms (SODA). (2008)
28. Arackaparambil, C., Brody, J., Chakrabarti, A.: Functional monitoring without monotonicity. In: Procs of the 36th ACM International Colloquium on Automata, Languages and Programming (ICALP). (2009)
29. Gibbons, P.B., Tirthapura, S.: Estimating simple functions on the union of data streams. In: Proceedings of the Thirteenth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA). (2001) 281–291
30. Haung, Z., Yi, K., Zhang, Q.: Randomized algorithms for tracking distributed count, frequencies and ranks. In: Proceedings of 31st ACM Symposium on Principles of Database Systems (PODS). (2012)
31. Z. Liu, B.R., Vojnovic, M.: Continuous distributed counting for non-monotonic streams. In: Proceedings of 31st ACM Symposium on Principles of Database Systems (PODS). (2012)
32. Yuan, J., Mills, K.: Monitoring the macroscopic effect of DDoS flooding attacks. IEEE Transactions on Dependable and Secure Computing **2**(4) (2005)
33. Basseville, M., Cardoso, J.F.: On entropies, divergences, and mean values. In: Proceedings of the IEEE International Symposium on Information Theory. (1995)
34. Ali, S.M., Silvey, S.D.: General Class of Coefficients of Divergence of One Distribution from Another. Journal of the Royal Statistical Society. Series B (Methodological) **28**(1) (1966) 131–142
35. Csiszár, I.: Information Measures: A Critical Survey. In: Transactions of the Seventh Prague Conference on Information Theory, Statistical Decision Functions, Random Processes, Dordrecht, D. Riedel (1978) 73–86

36. Morimoto, T.: Markov processes and the h-theorem. Journal of the Physical Society of Japan **18**(3) (1963) 328–331
37. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics **22**(1) (1951) 79–86
38. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by their probability distributions. Bulletin of the Calcutta Mathematical Society **35** (1943) 99–109
39. Muthukrishnan: Data Streams: Algorithms and Applications. Now Publishers Inc. (2005)
40. Anceaume, E., Busnel, Y., Rivetti, N.: Estimating the frequency of data items in massive distributed streams. In: Proceedings of the 4th IEEE Symposium on Network Cloud Computing and Applications (NCCA). (2015) 59–66
41. Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. J. Algorithms **55**(1) (2005) 58–75
42. the Internet Traffic Archive: http://ita.ee.lbl.gov/html/traces.html. Lawrence Berkeley National Laboratory (April 2008)
43. Bregman, L.M.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Computational Mathematics and Mathematical Physics **7**(3) (1967) 200–217
44. Hellinger, E.: Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. J. Reine Angew. Math. **136** (1909) 210–271
45. Csiszár, I.: Why least squares and maximum entropy? an axiomatic approach to inference for linear inverse problems. The Annals of Statistics **19**(4) (1991) 2032–2066
46. Amari, S.I., Cichocki, A.: Information geometry of divergence functions. Bulletin of the Polish Academy of Sciences: Technical Sciences **58**(1) (2010) 183–195
47. Amari, S.I.: α-Divergence Is Unique, Belonging to Both f-Divergence and Bregman Divergence Classes. IEEE Transactions on Information Theory **55**(11) (nov 2009) 4925–4931
48. Renyi, A.: On measures of information and entropy. In: Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability. (1960) 547–561

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

We have shown with Theorem 1, that the sketch codeviation matches exactly the codeviation if $k \geq |\operatorname{supp}(X) \cap \operatorname{supp}(Y)| + \mathbf{1}_{\operatorname{supp}(X) \setminus \operatorname{supp}(Y)} + \mathbf{1}_{\operatorname{supp}(Y) \setminus \operatorname{supp}(X)}$. 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 Ω-point distribution 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 2 [Upper bound of $\mathcal{E}_k(X, Y)$] Let $k \geq 1$ be the precision parameter of the sketch codeviation. For any two Ω-point distributions $X \in \mathcal{X}$ 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 15. Using Lemmata 16 and 17, we obtain the statement of the theorem. □

Lemma 15. *For any two Ω-point distributions $X \in \mathcal{X}$ 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:

$$\begin{cases} 0 \le x_i \le m_\mathcal{X} & \text{with } 1 \le i \le N, \\ 0 \le y_i \le m_\mathcal{Y} & \text{with } 1 \le i \le N, \\ \sum_{i=1}^{N} x_i = m_\mathcal{X}, \\ \sum_{i=1}^{N} y_i = m_\mathcal{Y}. \end{cases} \tag{11}$$

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

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

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

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

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 11, $\forall j \in \Omega \setminus \{i\}, x_j = 0$.
2. $y_N \le y_i$. 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 $\mathcal{E}_1 = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} x_i^\delta\, y_j^\delta = \frac{m_\mathcal{X} m_\mathcal{Y}}{N}$. □

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 16. *Let X_U and Y_U be two uniform Ω-point distributions,* 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 17. *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be any two Ω-point distributions. 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 Ω-point distributions, 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 Ω-point distributions. 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 Ω-point distributions 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 Ω-point distributions X and Y are uniform while the second part considers any two Ω-point distributions.

Case 1. Let X_U and Y_U be two uniform Ω-point distributions, *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

$$\mathcal{E}_k(\widetilde{X}_U, Y_U) = \min(E, E') \text{ with } \begin{cases} E = \min\limits_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a \in \rho, \ell, \ell' \in a}} \mathcal{E}_k^\rho(\widetilde{X}_U, Y_U) \\ \\ E' = \min\limits_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a, a' \in \rho, a \neq a' \\ \wedge \ell \in a \wedge \ell' \in a'}} \mathcal{E}_k^\rho(\widetilde{X}_U, Y_U). \end{cases} \tag{12}$$

Let us consider the first term E. We have

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

According to the second term E', we have

$$E' = \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a, a' \in \rho, a \neq a' \\ \wedge \ell \in a \wedge \ell' \in a'}} \left(\sum_{\substack{b \in \rho \\ \setminus \{a, a'\}}} \sum_{i \in b} \sum_{\substack{j \in b \\ \setminus \{i\}}} \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N^2} + \sum_{i \in a \setminus \{\ell\}} \sum_{j \in a \setminus \{i\}} \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N^2} \right.$$

$$\left. + \sum_{i \in a' \setminus \{\ell'\}} \sum_{j \in a' \setminus \{i\}} \frac{m_{\mathcal{X}} m_{\mathcal{Y}}}{N^2} + \sum_{j \in a \setminus \{\ell\}} \left(\frac{m_{\mathcal{X}}}{N} + 1 \right) \frac{m_{\mathcal{Y}}}{N} + \sum_{j \in a' \setminus \{\ell'\}} \left(\frac{m_{\mathcal{X}}}{N} - 1 \right) \frac{m_{\mathcal{Y}}}{N} \right)$$

$$= \min_{\substack{\rho \in \mathcal{P}_k(\Omega) \text{ s.t.} \\ \exists a, a' \in \rho, a \neq a' \\ \wedge \ell \in a \wedge \ell' \in a'}} \left(\mathcal{E}_k^{\rho}(X_U, Y_U) + \frac{m_{\mathcal{Y}}}{N} \left(|a| - |a'| \right) \right).$$

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 12. We get for the first term,

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

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

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 Ω-point distribution. Lemma 16 concludes the proof. □