HAL
archives-ouvertes.fr

An Information Divergence Estimation over Data Streams

Emmanuelle Anceaume, Yann Busnel

▶ **To cite this version:**

Emmanuelle Anceaume, Yann Busnel. An Information Divergence Estimation over Data Streams. 11th IEEE International Symposium on Network Computing and Applications (IEEE NCA12), Aug 2012, Cambridge, MA, United States. pp.Number 72. hal-00725097

HAL Id: hal-00725097
https://hal.archives-ouvertes.fr/hal-00725097

Submitted on 23 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Information Divergence Estimation over Data Streams

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

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

***Abstract*—In this paper, we consider the setting of large scale distributed systems, in which each node needs to quickly process a huge amount of data received in the form of a stream that may have been tampered with by an adversary. In this situation, a fundamental problem is how to detect and quantify the amount of work performed by the adversary. To address this issue, we have proposed in a prior work, AnKLe, a one pass algorithm for estimating the Kullback-Leibler divergence of an observed stream compared to the expected one. Experimental evaluations have shown that the estimation provided by AnKLe is accurate for different adversarial settings for which the quality of other methods dramatically decreases. In the present paper, considering n as the number of distinct data items in a stream, we show that AnKLe is an (ε, δ)-approximation algorithm with a space complexity $\tilde{\mathcal{O}}(\frac{1}{\varepsilon} + \frac{1}{\varepsilon^2})$ bits in "most" cases, and $\tilde{\mathcal{O}}(\frac{1}{\varepsilon} + \frac{n-\varepsilon^{-1}}{\varepsilon^2})$ otherwise. To the best of our knowledge, an approximation algorithm for estimating the Kullback-Leibler divergence has never been analyzed before.**

***Keywords*-Data stream; divergence; randomized approximation algorithm.**

I. Introduction

The main objective of this paper is the analysis of the quality of a one pass algorithm, AnKLe [1], in estimating the similarity between an observed data stream and the expected (*i.e.* idealized) one, in the context of massive data streams. This data may correspond to IP network traffic, sensors readings, nodes identifiers or any other data issued from distributed applications. In such contexts, nodes need to quickly process on the fly the flow of data. Moreover, nodes can only locally store very limited data and perform few operations on this data. Additionally, it is often the case that if some data has not been locally stored for further processing, once it has been read, it cannot be read anymore (this refers to the one-pass data streaming model). In this context, each node needs an efficient algorithm to process its input sequence. An algorithm is efficient [2] if it is capable of quickly processing a huge amount of data by using only $\text{poly}(1/\varepsilon, \log m, \log n)$ bits of memory, where ε is the approximation parameter of the function to be approximated, m the size of the input data stream and n the (unknown) number of distinct data items in the stream.

Given these constraint settings — a one-pass analysis of a huge amount of data with limited resources, both in space and time— AnKLe detects changes in the observed stream with respect to an expected behavior by relying on sampling techniques and information-theoretic methods. The metric used is the Kullback-Leibler (KL) divergence, which can be viewed as an extension of the Shannon entropy and is often referred to as the relative entropy [3].

In this paper, we analyze the quality of AnKLe in approximating the KL divergence between the expected stream and the observed one. An algorithm $\mathcal{A}$ is said to be an (ε, δ)-approximation of a function ϕ on σ if for any sequence of items in the input stream σ, $\mathcal{A}$ outputs $\hat{\phi}$ such that $\mathbb{P}\{\mid \hat{\phi} - \phi \mid > \varepsilon\phi\} < \delta$, where $\varepsilon, \delta > 0$ are given as parameters of the algorithm.

The paper is organized as follows. First, Section II reviews the related work on the estimation of the relative entropy of data streams while Section III describes the data stream model as well as the concepts of information theory that we intensively use in this work. Section IV briefly presents the different buildings blocks the AnKLe algorithm relies on, and finally Section V presents the analysis of this algorithm. Finally, we conclude in Section VII.

II. Related Work

In this paper, we consider the Kullback-Leibler (*i.e.*, the relative entropy) estimation problem. In information theory, the concept of entropy corresponds to the uncertainty of a random variable, and as a special case, the entropy of a stream quantifies the randomness of a data stream. On the other hand, relative entropy measures the difference between two distributions, and therefore the data stream relative entropy quantifies the amount of information separating one specific observed stream from expected ones.

Previous works have proposed efficient algorithms to accurately estimate the entropy of a data stream. Most of these works rely on the seminal algorithm designed by Alon, Matias and Szegedy [4]. Subsequently to this work, Guha *et al.* [5] have considered the entropy estimation problem in the random stream model, in which items are randomly distributed in the stream. Chakrabarti *et al.* [6] have studied the same problem but assuming the adversarial

stream model, that is, a stream in which the items are ordered according to an adversarial strategy. Furthermore, Chakrabarti *et al.* [6], [7] and Lall *et al.* [8] have considered the challenging issue of estimating the entropy accurately when the entropy is strictly less than one. Such streams have a few items with a high occurrence frequency while all the other items appear approximately with the same low frequency. In order to guarantee a small relative estimation error in this setting, one needs to decompose the analysis of the stream into two parts, one part keeping the highly frequent items and the other part comprising the items with the same low frequency. More details will be given in Section IV. A fundamental issue is to derive efficient one pass algorithms to estimate the relative entropy in presence of huge amount of data.

III. System Model and Background

A. Data stream model

We consider a system in which a node P receives a large data stream $\sigma = a_1, a_2, \ldots, a_m$, where the i-th element a_i of the stream is called an item. In the following, we describe a single instance of P, but clearly multiple instances of P may co-exist in a system (*e.g.*, in case P represents a router, a base station in a sensor network). The value u of an item is assumed to be drawn from a large universe N (*e.g.*, $|N| \sim 2^{32}$) and the length of the stream m can be very large too. Moreover, items can be repeated multiple times in the stream. The number of distinct items in the stream is denoted by n, and thus, we have $n < m$. We suppose that items arrive regularly and quickly, and due to memory constraints, need to be processed sequentially and in an online manner. Therefore, node P can locally store only a small fraction of the items and perform simple operations on them. The algorithms we consider in this work are characterized by the fact that they can approximate some function on σ with a very limited amount of memory. We refer the reader to [9] for a detailed description of data streaming models and algorithms.

B. Preliminaries

1) Entropy: Intuitively, the entropy is a measure of the randomness of a data stream σ. The entropy H_σ is minimum (*i.e.*, equal to zero) when all the items in the stream are the same, and it reaches its maximum (*i.e.*, equal to $\log m$)[1] when all the items in the stream are distinct. Specifically, we have

$$H_\sigma = -\sum_{u \in N} p_u \log p_u,$$

where $p_u = m_u/m$, for each $u \in N$, with $m_u = |\{j : a_j = u\}|$ representing the number of times the value u appears in the stream σ (by convention, $0 \log 0 = 0$). Without loss of generality, we assume that the items are ordered so that $m_1 \geq m_2 \geq \ldots \geq m_n$. Note that the number of times m_u item u appears in a stream is commonly called the frequency of item u. The norm of the entropy is defined as $F_H = \sum_{u \in N} m_u \log m_u$.

2) Kullback-Leibler divergence: The Kullback-Leibler (KL) divergence [10], also called the relative entropy, is a robust metric for measuring the statistical difference between two data streams. The KL divergence is a member of a larger class of distances known as the Ali-Silvey distances [11]. Given two probability distributions on events $p = \{p_1, \ldots, p_n\}$ and $q = \{q_1, \ldots, q_n\}$, the Kullback-Leibler divergence between p relative to q is defined as the expected value of the likelihood ratio with respect to q:

$$\mathcal{D}(p||q) = \sum_{u \in N} p_u \log \frac{p_u}{q_u} = H(p,q) - H(p),$$

where $H(p) = -\sum p_u \log p_u$ is the (empirical) entropy of p and $H(p,q) = -\sum p_u \log q_u$ is the cross entropy of p and q. As we use a logarithm in base 2, the divergence is measured in bits. When $p_n = q_n$, the KL divergence is minimal and is equal to zero. Let $p^{(\mathcal{U})}$ be the uniform distribution corresponding to a uniform stream (*i.e.*, $\forall u \in \sigma, p_u^{(\mathcal{U})} = \frac{1}{n}$), and q be the probability distribution corresponding to the input stream. In the rest of this paper and according to the classical use of the KL-divergence, we consider $\mathcal{D}(q||p^{(\mathcal{U})})$ as a measure of the divergence of the current stream from the ideal one. While all the distance measures in the Ali-Silvey distances are applicable to quantifying statistical differences between data streams, the KL divergence is particularly suited to our context since it gives rise to a small number of false positives when the two data streams are not significantly different.

3) Frequency moments: Frequency moments are important statistical tools that have been introduced by Alon *et al.* [4]. Computing frequency moments F_k allows to quantify the amount of skew in a data stream. For each $k \geq 0$, the k-th frequency moment F_k of σ is defined as $F_k = \sum_{u \in N} m_u^k$, where m_u represents the number of occurrences of u in the stream (*c.f.* the definition of m_u above). Among the remarkable moments, F_0 represents the number n of distinct elements in a stream while F_1 corresponds to the size m of the stream.

IV. The AnKLe algorithm

A. Building Blocks

In this section, we briefly describe three algorithms that form the building blocks of the AnKLe algorithm. All these algorithms have been designed in the stream data model (*cf.* Section III).

1) Estimating the k^{th} Moment of a Stream: The AnKLe algorithm is inspired from the method of Alon *et al.* [4] (called in the following the AMS algorithm), to approximate the KL divergence of a stream. The AMS algorithm estimates the k-th frequency moment of a stream as follows. It

[1] Thereafter, we will denote by log the logarithm in base 2.

computes a basic estimator which takes the form of a random variable X whose mean value is exactly equal to the k^{th} frequency moment of a stream and whose variance is very small. Specifically, X is defined as $X = m(r^k - (r-1)^k)$, where r is the exact number of times element v appears in the stream from a uniformly and randomly chosen position p (we have $a_p = v$) in the stream onwards. To improve the accuracy of the estimation, several independent basic estimators are computed on the stream (specifically $s_1 \times s_2$ basic estimators X_{ij}, for $1 \leq i \leq s_1$ and $1 \leq j \leq s_2$, for $s_1 \times s_2$ positions uniformly chosen at random in the stream σ), and the final estimator Y is set to be $Y = \mathrm{median}_{1\leq j\leq s_2}\left(\frac{1}{s_1}\sum_{i=1}^{s_1} X_{ij}\right)$.

Theorem 1 ([4]) *For any $\varepsilon, \delta \in (0,1)$, if $s_1 \geq Var[X]/(\varepsilon^2 E[X]^2)$ and $s_2 = 4\log(1/\delta)$, then Y is a (ε,δ)-approximation of $E[X]$,* i.e., $\mathbb{P}\{\mid E[X]-Y \mid > \varepsilon E[X]\} < \delta$.

2) Estimating the Number of Items in the Stream: The second algorithm due to Kane *et al.* [12] (referred to as the KNW algorithm in the following) computes an estimation $\hat{F}_0$ of the number of distinct items F_0 in a stream. The KNW algorithm builds upon the approaches proposed in [13] and [14] to optimally estimate F_0 both in space and update time. Briefly, the basic procedure consists in hashing all the received data items to a bit vector, so that each data item is mapped to bit i in the vector with probability $2^{-(i+1)}$. The returned value of the procedure is a function of r, where r is such that the r rightmost bits in the bit vector are all 0. To obtain a good estimator, the median value of k instances of the same procedure (using different hash functions) is returned.

Theorem 2 ([12]) *For any ε, their algorithm outputs $\hat{F}_0$ such that $\mathbb{P}\{|\hat{F}_0 - F_0| > \varepsilon\} < \delta$ where $\delta = 2/3$. The worst-case running time for each input symbol is $\mathcal{O}(1)$, and the total space required by the algorithm is $\mathcal{O}(1/\varepsilon^2 + \log n)$ bits, which makes this algorithm optimal.*

3) Determining Frequent Identifiers of a Stream: Misra and Gries [15] have proposed a deterministic algorithm that outputs items that occur more than $\frac{m}{k}$ in a stream. Their algorithm maintains k counters such that for each counter, its key is the item read from the stream and its value is related to the frequency of items. When an item is read from the stream, if that item has already a counter associated to it, then this counter is incremented. If this is not the case and if there are still free counters available, then one of these free counters is allocated to this new item and its value is set to 1. Otherwise, all the allocated counters are decremented by one, and if after this operation, some of them are equal to 0 then their keys are erased and the counters are released.

Theorem 3 ([15]) *The Misra and Gries [15] algorithm with parameter k returns for any data item j an estimate $\hat{m}_j$ such that $m_j - \frac{m}{k} \leq \hat{m}_j \leq m_j$ with $\mathcal{O}(k(\log m + \log n))$ bits of space.*

B. The AnKLe algorithm

For self-containment reasons, the pseudo-code of AnKLe is presented in Figure 1. Its principle stems from a rewriting of the KL divergence. From Definition 1, we have

$$\begin{aligned}\mathcal{D}(q_\sigma || p^{(\mathcal{U})}) &= \sum_{i=1}^{n} q_i \log(q_i) - \sum_{i=1}^{n} q_i \log\left(p_i^{(\mathcal{U})}\right) \\ &= \log(n) - \log(m) + \frac{1}{m}\sum_{i=1}^{n} m_i \log(m_i)\,. \end{aligned} \tag{1}$$

Thus estimating the KL-divergence amounts in *(i)* estimating the number of distinct items in the stream (*i.e.*, F_0) in order to obtain a good approximation of $\log(n)$, and *(ii)* estimating $\sum_{i=1}^{n} m_i \log(m_i)$, which corresponds to the norm of the entropy F_H. While the first point is solved by relying on the KNW [12] algorithm, the second point is tackled by extending the approach proposed by Alon *et al.* [4] to deal with arbitrary distributions of items in the input stream.

The pseudo-code of AnKLe consists of two phases, the first one (lines 3–11) is executed upon reception of the items of the stream, while the second one (lines 12–19) is run when m items have been read from the stream. The first phase is composed of three tasks (T_1, T_2 and T_3), executed in parallel. Task T_1 estimates the number of distinct items present in the stream, Task T_2 identifies the k most frequent items in the stream, and Task T_3 samples random items in the stream in order to compute their exact frequency. Specifically, Task T_3 (lines 8–11) consists in running a sampling estimator X on the stream. The basic estimator $X = X_{i,j}$ is designed so that its mean value is equal to the norm of the entropy F_H and its variance is small. More precisely, we have

$$X = m(r\log r - (r-1)\log(r-1)) \tag{2}$$

where r is the random variable representing the number of occurrence of an item ℓ in the stream. This item ℓ is such that its position j in the stream is a random number in $[m]$. The random variable r counts the number of times ℓ appears in the stream from position j onwards. Formally, r is defined as

$$r = \mid \{j : j \geq \ell, a_j = a_\ell\} \mid .$$

We can show as in [4], [8], that the basic estimator X is unbiased (*i.e.*, the expectation of X is equal to F_H). Specifically,

$$\begin{aligned} E[X] &= \frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{m_i} m(j\log j - (j-1)\log(j-1)) \\ &= \frac{m}{m}\sum_{i=1}^{n} m_i \log(m_i) \\ &= F_H. \end{aligned} \tag{3}$$

Input: An input stream σ of length m, c (number of counters in the Misra-Gries algorithm), s_1 and s_2 (for the size of the AMS-based matrix), k

Output: An estimation of $\mathcal{D}(q_\sigma||p^{(\mathcal{U})})$, the KL divergence between the observed stream and the uniform one

1 Choose $s_1 \times s_2$ random integers in $[1..m]$;
2 **for** $u_1 \in [0..s_1], u_2 \in [0..s_2]$ **do** $S[u_1, u_2] \leftarrow (\perp, \perp)$;
3 **for** $a_j \in \sigma$ **do**
4 $\quad v = a_j$;
5 $\quad$ **Task** T_1**:** $\hat{F}_0 \leftarrow$ KNW Algorithm (Algorithm [12]) fed with v;
6 $\quad$ **Task** T_2**:** $\hat{F} \leftarrow$ Misra-Gries Algorithm (Algorithm [15]) fed with v;
7 $\quad$ **Task** T_3**:**
8 $\quad$ **forall** entries i of matrix S such that $(s_i, r_i) \neq (\perp, \perp)$ **do**
9 $\quad\quad$ **if** $s_i = v$ **then** $r_i \leftarrow r_i + 1$;
10 $\quad$ **if** j is one the $s_1 \times s_2$ random integers **then**
11 $\quad\quad$ assign $(v, 1)$ to the first unused entry of S;
12 $\hat{F} \leftarrow$ the k most frequent items (s_i, r_i) of $\hat{F}$ and such that $r_i > e$;
13 **forall** entries i of matrix S **do**
14 $\quad$ **if** $(s_i, -) \in \hat{F}$ **then** $(s_i, r_i) \leftarrow (s_i, -)$;
15 $\quad$ **else** $(s_i, r_i) \leftarrow (s_i, m\,(r_i \log r_i - (r_i - 1)\log(r_i - 1))$;
16 $Y_S \leftarrow \text{median}_{1 \leq j \leq s_2} \left(\frac{1}{s_1} \sum_{i=1}^{s_1} S_{ij} \right)$;
17 $Y_{\hat{F}} \leftarrow \sum_{(s_i, r_i) \in \hat{F}} r_i \log r_i$;
18 $p \leftarrow 1 - \max\left(0, \dfrac{\min\left(Y_s, Y_{\hat{F}}\right) - m}{10 \cdot m}\right)$;
19 **return** $\hat{D} = \log \hat{F}_0 - \log m + \frac{p}{m}\left(Y_S + Y_{\hat{F}}\right)$;

Figure 1. AnKLe Algorithm

To improve the accuracy of the estimation, $s_1 \times s_2$ such basic estimators X_{ij} (for $1 \leq i \leq s_1$ and $1 \leq j \leq s_2$) are used, each one sampling a random position in the stream. Tracking these estimators consists in storing $s_1 \times s_2$ counters, each one counting the number of occurrences of an item whose position has been randomly chosen in the stream. Thus for each of these "tracked" items, an exact count of their frequency is continuously maintained starting from a random position in the stream.

The post-processing phase of AnKLe algorithm estimates the KL divergence of the input stream according to Relation (1). This phase is executed when m items have been read from the input stream. In this work, we suppose that m is a parameter of the algorithm, however by using techniques proposed in Chakrabarti *et al.* [7] we should be able to extend our solution to streams whose size is *a priori* unknown. To accurately estimate the KL divergence of the stream, one needs to cope with patterns in which a small number of items occur with a very high frequency with respect to the other items. When such patterns occur, the basic estimator X alone is unable to compute the norm of the entropy in bounded space [7]. Indeed, by analogy of the calculation performed in [4], the variance of the estimator grows with the norm of the entropy. Thus in presence of high frequency patterns, one needs to estimate the norm of the entropy using a different approach. AnKLe extends the solution proposed in Chakrabarti *et al.*, that consists in decomposing the computation of the entropy as the sum of the entropy of the k most frequent items and the estimation of the entropy of the remaining items of the stream.

Note that as previously mentioned, running the algorithm of Misra-Gries with k counters allows to output items that occur more than $\frac{m}{k}$ times in the stream of length m. In our case, we need a stronger property in the sense that we want to detect the k most frequent items. This can be achieved by increasing the number of counters maintained by the Misra-Gries algorithm so that if the frequency of any two items in the stream differs by at least εm_k then this is reflected in their estimated frequency value (see Section V).

Hence, the basic estimator X is computed on unfrequent items (*cf.*, lines 12–15) as done in Relation (2), while the contribution of the most frequent items on the norm of the entropy is directly computed as $\sum_{(s_i, r_i) \in \hat{F}} r_i \log r_i$ (*cf.*, line 17). Finally, to prevent some of the items to appear in both terms, we weight the contribution of both terms by p (*cf.*, line 18).

V. Analysis

In this section, we analyze the properties of the AnKLe algorithm given in Figure 1. This analysis is split into three phases. We first evaluate the quality of $Y_{\hat{F}}$ through Lemmatta 5, we then evaluate the quality of Y_S through Lemmatta 6 and 7, and finally derive the quality of AnKLe algorithm by combining the previous results with the one of [12] with Lemma 8.

In order to state the main theorem, we introduce the following notations: Let K be the set of the most frequent items i that satisfy $\hat{m}_i > e$ (if any) returned at line 12 in Figure 1. Let n_s and m_s be respectively defined as $n_s = n - |K|$ and $m_s = m - \sum_{k \in K} m_k$. Parameter n_s represents the number of the so-called "sparse" items (*i.e.*, the remaining items of the stream after having removed the most frequent ones as identified by Task T_2 and after the execution of line 12). In the same way, m_s represents the sub-stream of the original stream occupied by these sparse items. Finally, the norm of the entropy of this sub-stream is denoted by F_H^s.

Theorem 4 *For any δ and ε such that $1/3 < \delta < 1$ and $0 < \varepsilon < \frac{1}{2}$, and for any constant $\Delta > 0$, the AnKLe algorithm gives an (ε, δ)-approximation of the KL-divergence, using*

$$\mathcal{O}\left(\log n + \frac{1}{\varepsilon^2} + \left(\frac{1}{\varepsilon} + \frac{\mu}{\varepsilon^2}\log\frac{1}{\delta}\right)(\log n + \log m)\right)$$

bits of space where $\mu = (\log m + \log e - 1)$ if $F_H^s \geq \frac{m^2}{\Delta m_s}$, and $\mu = (n - \frac{1}{\varepsilon} - 1)$ otherwise.

In particular, taking Δ to be a constant, we have a poly-logarithmic space algorithm that works on streams whose F_H^s is not "too small". Note that this is the case for most of the streams, as Task T_2 aims to remove the most frequent items, raising then the norm of the entropy of the sparse sub-stream.

Proof: The first part of the proof is directly derived from Lemma 8. Regarding the space complexity of AnKLe, it is given by the sum of the complexity of each Task T_1, T_2 and T_3 added up with the space required for the post-processing phase, which is $\mathcal{O}(1)$. From respectively [12], [4], [15], we get that the space complexity of AnKLe is $\mathcal{O}\left(\mathcal{C}_{KNW} + \mathcal{C}_{MG} + \mathcal{C}_{AMS}\right)$, where:

$$\begin{cases} \mathcal{C}_{KNW} &= \mathcal{O}\left(\log n + \frac{1}{\varepsilon^2}\right) & [12] \\ \mathcal{C}_{MG} &= \mathcal{O}\left(c(\log n + \log m)\right) & [4] \\ \mathcal{C}_{AMS} &= \mathcal{O}\left(s_1 s_2(\log n + \log m)\right) & [15]. \end{cases}$$

Using Lemmata 5 and 6, and the hypothesis on c, s_1 and s_2 presented in Equation 5 (*cf.*, Section V-C), we obtain the statement of the theorem. ∎

We now show a series of results that prove lemmata 5, 6 and 8.

A. Evaluation of $Y_{\hat{F}}$

The following lemma computes the quality of $Y_{\hat{F}}$.

Lemma 5 *For any $\varepsilon > 0$, we have $\mathbb{P}\{|Y_F - Y_{\hat{F}}| > \varepsilon Y_F\} = 0$, where $Y_{\hat{F}}$ is defined at line 17 in Figure 1.*

Proof: We first show that by running the Misra-Gries algorithm [15] with c counters (instead of k, with $c > k$), we guarantee that the k most frequent items in the stream can be detected.

From Misra-Gries algorithm [15], we know that for any item $i \in [n]$ the estimated frequency $\hat{m}_i$ returned by the algorithm is lower or equal than the real one m_i. Moreover, the difference between m_i and $\hat{m}_i$ is no more than $\frac{m}{c}$ (*c.f.*, Theorem 3).

Now, let i and j be two items such that $m_i - m_j \geq \frac{m}{c}$. We have $\frac{m}{c} + m_j \leq m_i \leq \frac{m}{c} + \hat{m}_i$. Combined with $\hat{m}_j \leq m_j$, we get that $\hat{m}_i \geq \hat{m}_j$. As a consequence, if the number of counters c satisfies $c \geq \frac{2m}{\varepsilon m_k}$ then for any two items i and j such that $m_i \geq m_j + \frac{\varepsilon}{2} m_k$, we will be able to distinguish that $\hat{m}_i \geq \hat{m}_j$. Which proves the first part of the lemma. Now, by extracting the k items with the highest estimated frequency (among the c ones returned by the algorithm), we guarantee that these k items are the most frequent items in the stream. By convention (*cf.*, Section III-B), and by definition of K, any of these k most frequent items i (if they exist) belong to K and are such that $m_i \geq m_k$, where $m_k \geq e$.

The function $x \mapsto \frac{\log x}{x}$ is a decreasing function for any $x \geq e$. Thus, for any $i \in K$, $m_i \log \hat{m}_i \geq \hat{m}_i \log m_i$. Thus

$$\begin{aligned} & m_i \log m_i - \hat{m}_i \log \hat{m}_i \\ &= (m_i - \hat{m}_i)\log m_i + \hat{m}_i(\log m_i - \log \hat{m}_i) \\ &\leq (m_i - \hat{m}_i)\log m_i + (m_i - \hat{m}_i)\log \hat{m}_i \\ &\leq 2(m_i - \hat{m}_i)\log m_i \\ &\leq \varepsilon m_k \log m_i \\ &\leq \varepsilon m_i \log m_i. \end{aligned}$$

Finally,

$$\sum_{i \in K} m_i \log m_i - \sum_{i \in K} \hat{m}_i \log \hat{m}_i \leq \varepsilon \sum_{i \in K} m_i \log m_i.$$

Consequently, $\mathbb{P}\{|Y_F - Y_{\hat{F}}| > \varepsilon Y_F\} = 0$, which ends the proof of the Lemma. ∎

B. Evaluation of Y_s

Let X^s be the same estimator as X (which has been defined in Section IV-B), but X^s is defined only on sparse items in the stream. From a derivation similar to the one used in Relation (1), X^s is an unbiased estimator of F_H^s.

Lemma 6 *$\forall \varepsilon, \delta > 0$, it exists s_1 and s_2 such that*

$$\mathbb{P}\{|Y_s - F_H^s| > \varepsilon F_H^s\} < \delta.$$

Proof: By definition, $\{X_{i,j}\}_{i \in [s_1], j \in [s_2]}$ is a collection of independent random variables with each $X_{i,j}$ distributed

identically to X^s. We recall form the algorithm that

$$Y_s = \text{median}_{j \in s_2} \left(\frac{1}{s_1} \sum_{i \in [s_1]} X_{i,j} \right).$$

Given ε and δ, we show in the following that there exists a positive constant η such that if $s_1 = \frac{3V[X^s]}{\varepsilon^2 E[X^s]^2}$ and $s_2 = \eta \log \frac{1}{\delta}$, we have $\mathbb{P}\{|Y_s - F_H^s| > \varepsilon F_H^s\} < \delta$.

For each $j \in [s_2]$, consider $Y_j = \frac{1}{s_1} \sum_{i \in [s_1]} X_{i,j}$. Then, by linearity of expectation, we have $E[Y_j] = F_H^s$. Since the variables $Y_{i,j}$ are (at least) pairwise independent, we have

$$V[Y_j] = \frac{1}{s_1} \sum_{i \in [s_1]} V[X_{i,j}] = \frac{V[X^s]}{s_1}.$$

Applying Chebyshev's inequality, we have

$$\begin{aligned} \mathbb{P}\{|Y_j - F_H^s| > \varepsilon F_H^s\} &< \frac{V[Y_j]}{(\varepsilon F_H^s)^2} \\ &= \frac{V[X^s]}{s_1 \varepsilon^2 E[X^s]^2} = \frac{1}{3}. \end{aligned}$$

Given the fact that we run s_2 copies of this estimator Y_j in parallel, by a standard Chernoff bound, the probability that the median of these estimations exceeds $3X^s$ is $2^{-\Omega(s_2)}$. Similarly, the probability that the median is below $\frac{X^s}{3}$ is also $2^{-\Omega(s_2)}$. By choosing $s_2 = \Theta(\log \frac{1}{\delta})$, we can make the sum of these two probabilities work out at most δ. Then, it exists an appropriate choice of η that give us

$$\mathbb{P}\{|Y_s - F_H^s| > \varepsilon F_H^s\} < \delta$$

that concludes the proof. ■

We now derive a relation between $E[X^s]$ and $V[X^s]$ to deduce an estimation on the size of s_1.

Lemma 7 *Let us consider the sub-stream m_s populated by sparse items. Then, for any constant $\Delta > 0$ if $F_H^s \geq \frac{m^2}{\Delta m_s}$ then*

$$V[X^s] \leq \Delta(\log m + \log e - 1) E[X^s]^2$$

otherwise

$$V[X^s] \leq (n_s - 1) E[X^s]^2.$$

Proof: Consider the variance of X^s.

$$\begin{aligned} V[X^s] &= E[X^{s\,2}] - E[X^s]^2 \\ &= n_s \left(\sum_{i \in [n] \setminus K} \frac{n^2 m_i^2 \log^2 m_i}{n_s^2} \right) - E[X^s]^2 \\ &\leq n_s E[X^s]^2 - E[X^s]^2 \end{aligned}$$

as the sum of square is lower than the square of the sum. We can drastically improve this bound when the norm of the entropy of the sub-stream F_H^s is not too small, *i.e.*, when

$$F_H^s \geq \frac{m^2}{\Delta m_s}, \tag{4}$$

where Δ is positive constant. Let $g : x \mapsto x \log x$. Following the same approach as [6], we have:

$$\begin{aligned} E[X^{s\,2}] &= \frac{1}{m_s} \sum_{i=1}^{n_s} \sum_{j=1}^{m_i} m^2 (g(j) - g(j-1))^2 \\ &\leq \frac{m^2}{m_s} \cdot \max_{1 \leq j \leq m} \{g(j) - g(j-1)\} \cdot F_H^s \\ &= \frac{m^2}{m_s} \cdot \sup\{g'(x) : x \in (0; m]\} \cdot F_H^s \\ &\leq \frac{m^2}{m_s} (\log e + \log m) F_H^s \\ &\leq \Delta(\log e + \log m) {F_H^s}^2 \quad \text{(from Equation 4)} \\ &\leq \Delta(\log e + \log m) E[X^s]^2 \end{aligned}$$

that conclude the proof. ■

C. Evaluation of $\hat{D}$

As $|K| < 1/\varepsilon$, we are now able to explicitly give the value of all the parameters of tasks T_2 and T_3:

$$\begin{cases} c = \mathcal{O}\left(\frac{1}{\varepsilon}\right) \\ s_1 = \begin{cases} \mathcal{O}\left(\frac{\log m + \log e - 1}{\varepsilon^2}\right) & \text{if } F_H^s \geq \frac{m^2}{\Delta m_s} \\ \mathcal{O}\left(\frac{n_s - 1}{\varepsilon^2}\right) & \text{otherwise} \end{cases} \\ s_2 = \mathcal{O}\left(\log \frac{1}{\delta}\right) \end{cases} \tag{5}$$

Using these values, we have the necessary material to derive the global quality of AnKLe. First of all, by linearity of expectation, the random variable D is an unbiased estimator of $\mathcal{D}(q_\sigma || p^{(\mathcal{U})})$, given by Equation 1.

Lemma 8 *Given $\varepsilon > 0$ and $\delta > \frac{1}{3}$, we have*

$$\mathbb{P}\left\{ |\hat{D} - \mathcal{D}(q_\sigma || p^{(\mathcal{U})})| > \varepsilon \mathcal{D}(q_\sigma || p^{(\mathcal{U})}) \right\} < \delta.$$

Proof: Let $\delta' = \delta - \frac{1}{3}$. Then we have $\delta' > 0$.

Combining the independence of Y_s and $Y_{\hat{F}}$ and Lemmata 5 and 6, we have:

$$\begin{aligned} &\mathbb{P}\{|Y_s + Y_{\hat{F}} - E[Y_s + Y_{\hat{F}}]| > \varepsilon E[Y_s + Y_{\hat{F}}]\} \\ &\quad \leq \mathbb{P}\{|Y_s - E[Y_s]| > \varepsilon E[Y_s]\} \\ &\qquad + \mathbb{P}\{|Y_{\hat{F}} - E[Y_{\hat{F}}]| > \varepsilon E[Y_{\hat{F}}]\} \\ &\quad < \delta' \end{aligned}$$

By definition of $\mathcal{D}(q_\sigma || p^{(\mathcal{U})})$ in Relation 1 and $\hat{D}$ in

AnKLe algorithm at line 19, we have :

$$\begin{aligned}
&\mathbb{P}\{|\hat{D} - \mathcal{D}(q_\sigma||p^{(\mathcal{U})})| > \varepsilon\mathcal{D}(q_\sigma||p^{(\mathcal{U})})]\} \\
&\quad\leq \quad \mathbb{P}\{|\log\hat{F}_0 - \log F_0| > \varepsilon\log F_0\} \\
&\qquad + \mathbb{P}\left\{\frac{1}{m}|\hat{F_H} - F_H| > \frac{\varepsilon}{m}E[F_H]\right\} \\
&\quad\leq \quad \frac{1}{3} + \mathbb{P}\left\{\frac{1}{m}|\hat{F_H} - F_H| > \frac{\varepsilon}{m}E[F_H]\right\} \quad \text{(from [12])} \\
&\quad\leq \quad \frac{1}{3} + \mathbb{P}\left\{|Y_s + Y_{\hat{F}} - E[Y_s + Y_{\hat{F}}]| > \varepsilon E[Y_s + Y_{\hat{F}}]\right\} \\
&\quad< \quad \frac{1}{3} + \delta' = \delta
\end{aligned}$$

that concludes the proof. ■

VI. Performance evaluation

Intensive executions of AnKLe have been presented in [1]. In this paper, we do not recall all these results but summarize them.

The accuracy of AnKLe has been evaluated by comparing its estimation of the KL divergence with the exact value of the KL divergence computed between an observed input stream and an uniform one. We have also compared AnKLe to adapted versions of the estimator-based algorithms of Alon *et al.* [4] and Chakrabarti *et al.* [7]. In the former case, the original estimator computes the k-th frequency moment of a stream, while in the latter case, the original estimator measures the entropy of a stream. In both cases, we have adapted both algorithms to compute instead the norm of the entropy.

All the experiments have been conducted on synthetic traces of streams whose distributions are: Uniform, Zipf (*aka* power law) with parameter $\alpha \in \{1, 2, 4\}$, Binomial and Pascal (*aka* Negative Binomial).

Figure VI summarizes the results obtained for the AnKLe, AMS and CCM estimators, averaged over 45,000 experiments (*i.e.* 750 different settings with 10 repetitions for each setting, over 6 distributions). For clarity of the Figure, the average value of CCM for Zipf with $\alpha = 1$ has been cropped, as the estimated value of the KL divergence by CCM is around 8.3.

These results clearly show that AnKLe outperforms the estimator CCM for all the distributions, even in scenario in which CCM should excel (*i.e.*, Zipf with $\alpha = 4$), as this corresponds to a stream in which a very frequent item exists in the observed stream. Compared with the AMS estimator, the results obtained with AnKLe are for most of them better except for the Zipf distribution with $\alpha = 2$. But even for this specific distribution, the standard deviation of AnKLe is four times smaller than the one of AMS (*i.e.*, 0.09 versus 0.36), thus demonstrating that AnKLe provides a more robust and stable estimation than AMS on this distribution.

VII. Conclusion and future works

In this paper, we have proposed the analysis of AnKLe, a randomized algorithm for estimating the KL divergence between the observed stream and the expected one. As initially raised in [1], AnKLe requires a single pass over the data stream to estimate the KL divergence. In this paper, we characterize how the different parameters impact the precision of the estimation and the space complexity of AnKLe (and *vice-versa*). We have shown that AnKLe is an (ε, δ)-approximation algorithm with a space complexity $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2} + \frac{1}{\varepsilon}\right)$ bits in "most" of the cases.

While we have supposed so far that the length of the stream m is a parameter that has to be fixed in advance, we left as future work the design of an extension of the algorithm for which the length is not specified in advance by using windowing techniques as the one proposed by Chakrabarti *et al.* [7].

References

[1] E. Anceaume, Y. Busnel, and S. Gambs, "AnKLe: detecting attacks in large scale systems via information divergence," in *Proceedings of the Ninth European Dependable Computing Conference (EDCC)*, 2012, to appear – Preprint: http://tinyurl.com/ABG-EDCC12.

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

[3] T. Cover and J. Thomas, "Elements of information theory," *Wiley New York*, 1991.

[4] N. Alon, Y. Matias, and M. Szegedy, "The space complexity of approximating the frequency moments," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (STOC)*, 1996, pp. 20–29.

[5] S. Guha, A. McGregor, and S. Venkatasubramanian, "Streaming and sublinear approximation of entropy and information distances," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006, pp. 733–742.

[6] A. Chakrabarti, K. D. Ba, and S. Muthukrishnan, "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.

[7] A. Chakrabarti, G. Cormode, and A. McGregor, "A near-optimal algorithm for computing the entropy of a stream," in *In ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 328–335.

[8] A. Lall, V. Sekar, M. Ogihara, J. Xu, and H. Zhang, "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.

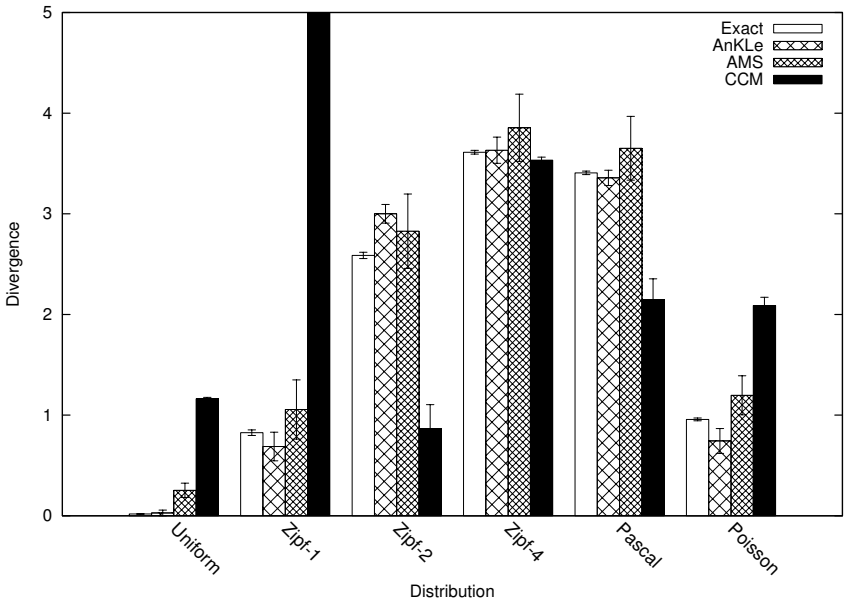

Figure 2. KL divergence estimation as a function of k

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

[10] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. [Online]. Available: http://dx.doi.org/10.2307/2236703

[11] S. M. Ali and S. D. Silvey, "General Class of Coefficients of Divergence of One Distribution from Another," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 28, no. 1, pp. 131–142, 1966.

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

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

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

[15] J. Misra and D. Gries, "Finding repeated elements," *Science of Computer Programming*, vol. 2, no. 2, pp. 143–152, 1982.