AnKLe: détection automatique d'attaques par divergence d'information

Emmanuelle Anceaume, Yann Busnel, Sébastien Gambs

To cite this version:

Emmanuelle Anceaume, Yann Busnel, Sébastien Gambs. AnKLe: détection automatique d'attaques par divergence d'information. Mathieu, Fabien et Hanusse, Nicolas. 14èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel), 2012, La Grande Motte, France. pp.1-4, 2012. <hal-00688352>

HAL Id: hal-00688352

https://hal.archives-ouvertes.fr/hal-00688352

Submitted on 17 Apr 2012

AnKLe: détection automatique d'attaques par divergence d'information

Emmanuelle Anceaume[1] and Yann Busnel[2] and Sébastien Gambs[3]

[1]*IRISA & CNRS, Campus Universitaire de Beaulieu, F-35042 Rennes Cedex, France*
[2]*LINA & Université de Nantes, 2 rue de la Houssinière, BP 92208, 44322 Nantes Cedex 03, France*
[3]*IRISA & Université de Rennes 1 - INRIA, Campus Universitaire de Beaulieu, F-35042 Rennes Cedex, France*

Dans cet article, nous considérons le contexte de très grands systèmes distribués, au sein desquels chaque nœud doit pouvoir rapidement analyser une grande quantité d'information, lui arrivant sous la forme d'un flux. Ce dernier ayant pu être modifié par un adversaire, un problème fondamental consiste en la détection et la quantification d'actions malveillantes effectuées sur ce flux. Dans ce but, nous proposons AnKLe (pour Attack-tolerant eNhanced Kullback-Leibler divergence Estimator), un algorithme local permettant d'estimer la divergence de Kullback-Leibler entre un flux observé et le flux espéré. AnKLe combine des techniques d'échantillonnage et des méthodes de théorie de l'information. Il est efficace à la fois en complexité en terme d'espace et en temps, et ne nécessite qu'une passe unique sur le flux. Les résultats expérimentaux montre que l'estimateur fourni par AnKLe est pertinent pour plusieurs types d'attaques, pour lesquels les autres méthodes existantes sont significativement moins performantes.

Keywords: Flux de données, Divergence de Kullback-Leibler, Echantillonnage, Adversaire Byzantin.

1 Introduction

The main objective of this paper is to propose an algorithm for estimating the similarity between an observed data stream and the expected (*i.e.* idealized) one in the context of massive data streams. More precisely, we consider the setting of large scale distributed systems, in which each node needs to quickly process a huge amount of data on the fly. Typically, this data corresponds to IP network traffic, sensors readings, nodes identifiers or any other data issued from distributed applications. 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).

Given our constraint settings, we propose an algorithm to detect changes in the observed stream with respect to an expected behavior by relying on sampling techniques and information-theoretic methods. More precisely, by adequately sampling the observed data stream, we estimate with high accuracy the distance between the expected stream and the observed one, and this even if the stream has been tampered with by an adversary. The metric, we use in our context 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. Our main contribution is the proposition of AnKLe (Attack-tolerant eNhanced Kullback-Leibler divergence Estimator), an algorithm that estimates the relative entropy between the observed stream and the expected ones in the context of massive data streams, while using only a memory of small size to cope with the very strict space constraint. Extensive simulations indicate that while AnKLe relies on sampling techniques, the accuracy of the estimation is very high. Finally, AnKLe is versatile enough to cope with any type of input distribution, including distribution that have been generated by an adversary. To the best of our knowledge, an algorithm combining all these strengths for the estimation of relative entropy has never been published before in the literature. Due to space constraints, we are not able to describe the related work, nevertheless, we refer the reader to [ABG12] for the full details.

2 System Model and Background

System 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. The value u of an item is assumed to be drawn from a large universe N and the length of the stream m is very high (*e.g.*, 2^{32}). 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 $n \leq m$. We suppose that items arrive on a regular basis 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.

Adversary Model. We suppose that the adversary is omnipotent in the sense that it may actively tamper with the data stream of any node by observing, inserting, dropping or re-ordering items of their input stream. The activity of the adversary can be detected by an honest node provided that it can accurately estimate the divergence between the observed stream and the ideal one. The presence of a high level of divergence is important as it may be a good indicator of attacks.

Preliminaries. *Kullback-Leibler divergence.* The Kullback-Leibler (KL) divergence, also called the relative entropy, is a robust metric for measuring the statistical difference between two data streams. 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_u relative to q_u is defined as a likelihood ratio with respect to q_u : $\mathcal{D}(p||q) = \sum_{u \in N} p_u \log \frac{p_u}{q_u} = H(p,q) - H(p)$, where $H(p) = -\sum p_u \log p_u$ is the empirical (*i.e.*, Shannon) entropy of p and $H(p,q) = -\sum p_u \log q_u$ is the cross entropy.

Frequency moments. Frequency moments F_ℓ are important statistical tools, which allows to quantify the amount of skew in a data stream. Among the remarkable moments, F_0 represents the number of distinct elements in a stream (*i.e.*, n in our model), while F_1 corresponds to the size m of the stream. For each $\ell \geq 0$, the ℓ-th frequency moment F_ℓ of σ is defined as $F_\ell = \sum_{u \in N} (m_u)^\ell$, where m_u represents the number of times the value u appears in the stream σ.

3 Detecting Adversarial Behaviors via KL Divergence Estimation

Building Blocks. We briefly review three algorithms that form the building blocks of the AnKLe algorithm. The first one, due to Alon *et al.* [AMS96] estimates the ℓ-th frequency moment of a stream. Although we do not need such a quantity, we adopt the structure of their algorithm to estimate the relative entropy of a stream. The second algorithm due to Bar-Yossef *et al.* [BYJK$^+$02] estimates the number of distinct items in a stream (*i.e.*, n). Finally the third algorithm, proposed by Misra and Gries [MG82], estimates the k most frequent items of a stream. Details about this algorithm are provided in [ABG12].

The AnKLe algorithm. This section describes AnKLe, the algorithm we propose for computing the KL divergence of a stream. Our starting point is the re-writing of the KL divergence as follows :

$$\mathcal{D}(q_\sigma||p^{(\mathcal{U})}) = \frac{1}{m}\left(\sum_{u \in N} m_u \log\left(\frac{m_u}{m}\right) - \sum_{u \in N} m_u \log\left(\frac{1}{n}\right)\right) = \log(n) - \log(m) + \frac{1}{m}\sum_{u \in N} m_u \log(m_u). \quad (1)$$

Thus estimating the KL-divergence amounts in estimating the number of distinct items in the stream in order to obtain a good approximation of $\log(n)$, and in estimating the norm of the entropy F_H. While the first point is solved by relying on BJKST algorithm [BYJK$^+$02], the second point is tackled by extending the approach proposed by Alon et al [AMS96] to deal with arbitrary distributions of items in the input stream.

Algorithm 1 presents AnKLe, which consists of two phases for computing the KL divergence : the first one is executed upon reception of the items of the stream ("the for loop $a_j \in \sigma$"), while the second one is run when m items have been read from the stream ("the forall loop until end"). The first phase is composed of three tasks, which are 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 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 $X = m(r \log r - (r-1)\log(r-1))$

where r is the random variable representing the number of occurrence of an item v, which is chosen at random in the stream through its position j being randomly generated (see the first line). The random variable r counts the number of times v appears in the stream from position j onwards. Formally, r is defined as $r = \mid \{j : j \geq v, a_j = a_v\} \mid$. We can show as in [AMS96], that the basic estimator X is unbiased (*i.e.*, X's expectation is equal to $F_H) : E[X] = \frac{1}{m}\sum_{u\in N}\sum_{j=1}^{m_u} m(j\log j-(j-1)\log(j-1)) = \frac{m}{m}\sum_{u\in N} m_u \log(m_u) = F_H$.

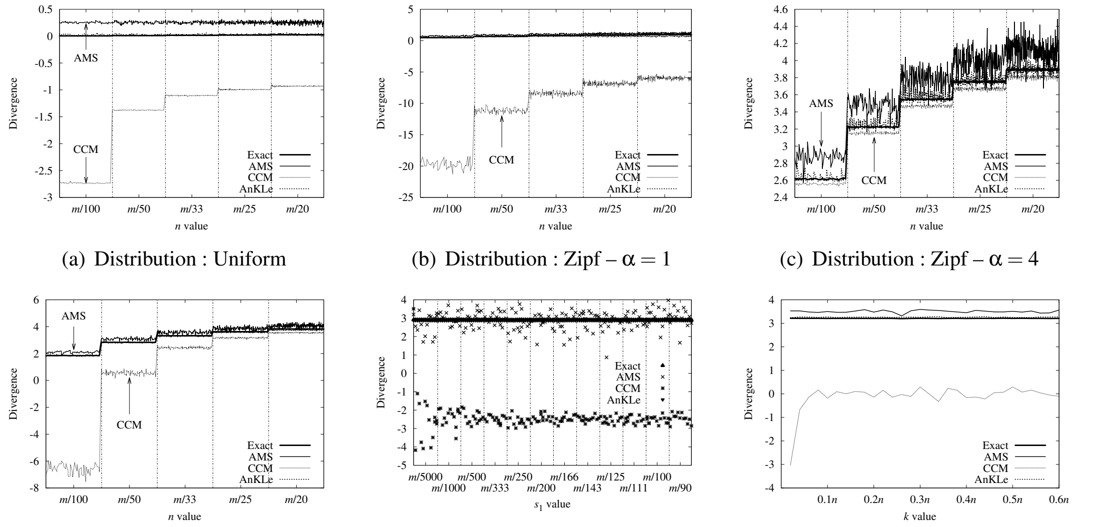
To improve the accuracy of the estimation, $s_1 \times s_2$ such basic estimators X_{ij} (for $1 \leq i \leq s_1$ and $1 \leq j \leq s_2$) are used, each one sampling a random position in the stream. From the implementation point of view, tracking these estimators consists in storing $s_1 \times s_2$ counters, each one counting the number of occurrences of an item whose position has been randomly chosen in the stream. When item u is read from the input stream, if u has already one or more counters assigned to it then all these counters are incremented. In addition, if the position at which u has been read in the stream is one of the chosen locations, then a new counter is assigned to u, and its value is set to one. Thus for each of these "tracked" items, an exact count of their frequency is continuously maintained, starting from a random position in the stream.

Algorithm 1: AnKLe algorithm

```
Input: An input stream $\sigma$ of length $m$, $k$ (number of
       counters in the Misra-Gries algorithm), $s_1$ and
       $s_2$ (for the size of the AMS matrix)
Output: An estimation of $\mathcal{D}(q_\sigma || p^{(\mathcal{U})})$
Choose $s_1 \times s_2$ random integers in $[1..m]$;
for $u_1 \in [0..s_1], u_2 \in [0..s_2]$ do $S[u_1,u_2] \leftarrow (\perp,\perp)$;
for $a_j \in \sigma$ do
    $v = a_j$;
    Task $T_1$ :
    $\hat{F}_0 \leftarrow$ BJKST Algorithm [BYJK+02] fed with $v$;
    Task $T_2$ :
    $\hat{F} \leftarrow$ Misra-Gries Algorithm [MG82] fed with $v$;
    Task $T_3$ :
    forall the entries $i$ of $S$ st $(s_i,r_i) \neq (\perp,\perp)$ do
        if $s_i = v$ then
            $r_i \leftarrow r_i + 1$
    if $j$ is one the $s_1 \times s_2$ random integers then
        assign $(v,1)$ to the first unused entry of $S$
forall the entries $i$ of matrix $S$ do
    if $(s_i,-) \in \hat{F}$ then
        $X_i \leftarrow 0$            // $s_i$ is one of the frequent items
    else
        $X_i \leftarrow m(r_i \log r_i - (r_i - 1)\log(r_i - 1))$
$Y_S \leftarrow$ average of all non null entries $X_i$;
$Y_{\hat{F}} \leftarrow \sum_{(s_i,r_i)\in\hat{F}} r_i \log r_i$;
$p \leftarrow 1 - \max\left(0, \frac{\min(Y_S,Y_{\hat{F}})-m}{10\cdot m}\right)$;
return $D = \log\hat{F}_0 - \log m + \frac{p}{m}\left(Y_S + Y_{\hat{F}}\right)$;
```

The post-processing phase of AnKLe algorithm estimates the KL divergence of the input stream according to Relation (1). 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. In this case, the basic estimator X alone is unable to compute the norm of the entropy in bounded space as the variance of the estimator grows with the norm of the entropy. In Chakrabarti *et al.* [CCM07], the authors propose to decompose the computation of the entropy as the sum of the entropy of the most frequent items and the estimation of the entropy of the remaining items of the stream. In AnKLe, we extend their method to deal with any stream distribution in order to guarantee that whatever the strategy of the adversary, the error on the estimation is kept small (see for instance the performance analysis below). Therefore, the basic estimator X is computed on unfrequent items, while the contribution of highly 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.*, $Y_{\hat{F}}$), in which the set $\hat{F}$ represents the set of highly frequent items dynamically computed in Task T_2. Finally, to prevent some of the items to appear in both terms, we weight the contribution of both terms by p.

Performance Analysis. In this section, we evaluate the accuracy of AnKLe by comparing its estimation with the exact value of the KL divergence computed between the observed input stream and the uniform one. We also compare AnKLe to adapted versions of the estimator-based algorithms of Alon *et al.* [AMS96] and Chakrabarti *et al.* [CCM07]. 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, the adapted versions compute instead the norm of the entropy. All the experiments have been conducted on synthetic traces of streams generated from classical distributions that capture different adversarial strategies : Uniform, Poisson, Zipf, Pascal. More precisely, all the generated streams have a length of $m = 200,000$ items. We have tested 750 different settings of n, s_1, and s_2 (s_1 and s_2 being related to the size of the es-



(a) Distribution : Uniform

(b) Distribution : Zipf – $\alpha = 1$

(c) Distribution : Zipf – $\alpha = 4$

(d) Distribution : Pascal

(e) Estimation only as a function of s_1 and s_2 (Pascal Distribution). Settings : $n = m/125 - k = 0.1n$

(f) Estimation only as a function of k (Pascal Distribution). Settings : $n = m/100$ and $s1 = s2 = m/80$

FIGURE 1: KL divergence estimation as a function of n, k, s_1 and s_2

timator matrix in Task T_3), and k, the number of counters used in Task T_2. For each setting of parameters, we have conducted 10 trials of the same experiment and compute the average and the standard deviation. Results obtained for the AnKLe, AMS and CCM estimators, averaged over 45,000 experiments, clearly show that AnKLe outperforms other ones. Figure 1 shows the evolution of the KL divergence estimation as a function of n, k, s_1 and s_2. In the four first figures, the abscissa represents the number of distinct items in the stream as a ratio of its length m. For each value of $n \in \{m/100, \ldots, m/20\}$, all the other parameters k, s_1 et s_2 also vary in the experiments, respectively in $\{0.1n, \ldots, n\}$, {m/100,. . .,m/20}$, and \{m/100, \ldots, m/20\}$. Fig. 1(e) (resp. Fig. 1(f)) shows the influence of s_1 and s_2 (respectively of k) on KL divergence estimation. The main observation drawn from these figures is that AnKLe fully overlaps with the exact value of the KL divergence, which clearly demonstrates the robustness of this estimator in presence of any input streams.

Conclusion. To summarize, experiments have validated the high accuracy and robustness of AnKLe in presence of a very large spectrum of distributions. This illustrates the importance of the weighting factor applied to both terms of the estimator. We left as future work the theoretical analysis of the behavior of AnKLe. In particular, we would like to conduct a detailed analysis on how the different parameters impact the precision of the estimation and the space complexity of AnKLe (and *vice-versa*).

Références

[ABG12] E. Anceaume, Y. Busnel, and S. Gambs. AnKLe : Detecting Attacks in Large Scale Systems via Information Divergence. In *9th European Dependable Computing Conf. (EDCC)*, 2012.

[AMS96] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *28th ACM Symposium on Theory of computing (STOC)*, pages 20–29, 1996.

[BYJK$^+$02] Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan. Counting distinct elements in a data stream. In *6th Intl Workshop on Randomization and Approximation Techniques (RANDOM)*, pages 1–10, 2002.

[CCM07] A. Chakrabarti, G. Cormode, and A. McGregor. A near-optimal algorithm for computing the entropy of a stream. In *ACM-SIAM Symp. on Discrete Algorithms*, pages 328–335, 2007.

[MG82] J. Misra and D. Gries. Finding repeated elements. *Science of Computer Programming*, 2(2) :143–152, 1982.