

Early Video Content Popularity Detection with Change Point Analysis

Sotiris Skaperas
Department of Applied Informatics
University of Macedonia
Thessaloniki, Greece
sotskap@uom.edu.gr

Lefteris Mamas
Department of Applied Informatics
University of Macedonia
Thessaloniki, Greece
emamas@uom.edu.gr

Arsenia Chorti
ENSEA, UMR 8051, CNRS
Université Cergy-Pointoise
Cergy-Pontoise, France
arsenia.chorti@ensea.fr

Abstract—Video content is responsible for more than 70% of the global IP traffic. Consequently, it is important for content delivery infrastructures to rapidly detect and respond to changes in content popularity dynamics. For flexible and highly adaptive solutions, the capability for a quick response should be driven from early (real-time) and low-complexity content popularity detection schemes. In this paper, we focus on the early and low-complexity detection of video content popularity, which we address as a statistical change point (CP) detection problem. Our proposed methodology estimates in real-time the existence, the number, the magnitude and the direction of changes in the average number of video visits by combining: (i) off-line and on-line CP schemes; (ii) an improved measurements window segmentation heuristic for the detection of multiple CPs; and (iii) a variation of the moving average convergence divergence (MACD) indicator to detect the direction of changes. We evaluated the proposed framework using a large database of real youtube video visits. The proposed algorithm is shown to accurately identify CPs and the direction of change in the off-line phase. Finally, a few illustrative examples of two variations of the on-line algorithm are also included.

Index Terms—video content popularity detection, change point analysis, time-series segmentation

I. INTRODUCTION

Video content is projected to account for 82% of the global Internet traffic by 2020, significantly increased from 72% in 2016 [1]. This highlights the importance of early detection of changes in content popularity which impacts the network traffic and server utilization. This problem requires a fresh view, since new networking and cloud paradigms appeared lately with significant elasticity capabilities, including software defined networking, new cloud orchestration proposals [2] and evolutions in the content distribution networks (CDNs) [3]. Such approaches offer new means to respond quickly to the changes in the content popularity dynamics, with the appropriate adaptations, e.g., efficient capacity and server resource allocation, load balancing or content caching.

Content popularity prediction is challenging, because many different factors from both the digital and physical world influence the Internet surfing behavior of the users [4], including (i) the quality, type (e.g., commercial or user-provided) and lifetime of content; (ii) its relevance to users and physical events; (iii) the social interactions between users; and (iv) the content promotion strategies involved. Furthermore, mid-term and

long-term prediction – and corresponding adaptations in the network or cloud environment – can prove inaccurate and thus result in sub-optimal service planning and provisioning, utilization of resources or violation of service level agreements. In this work, we revisit content popularity prediction in the context of modern, flexible networking and cloud approaches, that are highly adaptable and can respond to short-term dynamics. With accurate, on-line content popularity detection, discrepancies between predictions and actual changes can be alleviated. Our proposed approach is real-time, lightweight, accurate and accounts for historical information on content visits.

In our analysis we propose the employment of change point (CP) detection theory and algorithms; their suitability is confirmed by analyzing a large number of real youtube video data. In this contribution, the early content popularity detection problem is addressed with a novel CP detection methodology, consisting of a training phase using historical data and an on-line CP detection phase. In the training phase, we employ modified off-line CP detection schemes to configure a set of parameters that we later use in the sequential (on-line) algorithm. This approach is shown to greatly improve the accuracy of the on-line detector, as in essence the algorithm parametrization is not arbitrary but rather extracted from corresponding historical data. To the best of our knowledge, it is the first time in the literature that sequential (on-line) and retrospective (off-line) CP detection algorithms are combined in a single algorithm.

A second novelty in our approach is that we complement the off-line scheme with a novel, more accurate segmentation algorithm used for the detection of multiple CPs. Finally, we enrich the on-line algorithm with a modified exponential moving average filter that allows us to detect the direction of changes and accordingly respond in terms of load balancing. We give representative examples from two approaches of on-line algorithms. As shown in section III, the proposed overall CP detector is based on sufficiently general and convenient assumptions. Moreover, unlike other approaches, we employ methods that allow dependence between observations, leading to more realistic assumptions for the statistical structure of the content visits.

The rest of the paper is organized as follows. In Section II

we contrast our approach to related works. In Section III we discuss our methodological approach and proposed schemes. Section IV includes a performance analysis of our algorithm using a database of real youtube video views. In Section V, we discuss our future plans and conclude the paper.

II. RELATED WORKS

The area of content popularity attracted a lot of interest in recent years, because of its importance in a number of areas, such as network dimensioning (e.g., capacity planning or scaling up of resources), on-line marketing (e.g., advertising, recommendation systems) or real-world outcome prediction (e.g., analysis of economical trends) [4]. Most of content popularity studies relevant to our work consider the aggregate behavior of a particular content, focus on content popularity prediction and adopt parametric approaches. They can be categorized as: (i) cumulative growth studies, estimating the "amount of attention" from the publication instance to the prediction moment [5]; (ii) temporal analysis approaches, i.e., how content visits evolve over time [6]; and (iii) clustering methods of content with similar popularity trends [7], [8]. Finally, statistical characterizations of the distribution of visits, show that the underlying distributions are either Zipf [9] or Zipf-Mandelbrot [10] for both commercial or user-generated content.

In contrast to existing studies, we introduce a novel CP detection methodology that provides accurate, lightweight, and early, on-line CP detection of content popularity. We formulate the detection of a change in content popularity as a statistical hypothesis test and deploy non-parametric procedures to avoid a particular distribution assumption (such as a specific copula model). This context ensures low convergence time, since it avoids estimating a large number of model parameters and restrictive assumptions that may not match the time-series structure. Furthermore, we avoid problems of parametric models that require parameters' fitting and selection, which may be inapplicable as new data becomes available.

Up to now there are only a handful of proposals addressing the challenges of new flexible networking and cloud architectures accounting for content popularity. Exceptions include [11] in which a logistic-loss machine-learning approach to content popularity prediction is applied for a Fog RAN environment, and, our recent paper [3]. In [3], the algorithm described here is integrated in an elastic CDN framework based on lightweight cloud capabilities using Unikernels. [3] focuses on the platform details rather than on the CP algorithm; it confirms experimentally the suitability of the latter for relevant flexible network and cloud architectures. The first detailed description of the enhanced CP detection algorithm is presented in the following.

III. EARLY CP DETECTION ALGORITHM

In this section, we present our on-line content popularity detection algorithm. The basic idea is to detect, in real-time, changes in the average number of content popularity views.

For this reason, we developed an iterative algorithm that consists of: (i) a training phase based on off-line CP detection – during this phase the algorithm parameters are determined; and (ii) a sequential phase based on on-line CP detection that identifies in real-time qualitative changes in the content popularity dynamics. We elaborate on the two parts of the proposed algorithm below and then provide the corresponding pseudo-code.

A. Training (off-line) Phase

Here, we discuss the training phase of the algorithm. We begin by analyzing the foundational aspects of the off-line scheme that is used to detect a CP in past (historical) visit measurements. Standard off-line schemes can only detect one CP. To address the issue of detection of multiple CPs, we enrich the basic scheme with a novel time-series segmentation heuristic, that belongs to the family of binary segmentation algorithms.

1) *Basic Off-line Approach:* Let $\{X_n : n \in \mathbb{N}\}$ be a sequence of d -dimensional random vectors (r.v.), representing the number of video views within a time period for a specific video, taking values in \mathbb{N}^d . We assume that X_1, \dots, X_N can be written as,

$$X_n = \mu_n + Y_n, \quad 1 \leq n \leq N \quad (1)$$

where $\{\mu_n : n \in \mathbb{N}\}$ is the mean value of video visits and $\{Y_n : n \in \mathbb{N}\}$ a random component, with zero mean, $E[Y_n] = 0$, and positive definite covariance matrix, $E[Y_n Y_n^T] = \Sigma$. We further assume that it is m -dependent, implying that for $l, s, m \in \mathbb{N}$, Y_l is independent of Y_s if $|l - s| > m$.

The off-line analysis tests the constancy (or not) of the mean values up to the current time N . Hence, we define the following null hypothesis of constant means,

$$H_0 : \mu_1 = \dots = \mu_N$$

against the alternative,

$$H_1 : \mu_1 = \dots = \mu_k \neq \mu_{k+1} = \dots = \mu_N$$

indicating that the mean value changed at the unknown (time) point $k \in \{1, N\}$.

Considering (1) and the corresponding assumptions for the stochastic process X_n , we develop a non-parametric CUSUM test statistic following [12]. The test statistic can be viewed as a max-type procedure,

$$M = \max_{1 \leq n \leq N} C_n^T \widehat{\Omega}_N^{-1} C_n \quad (2)$$

where the parameter C_n is the typical CUSUM,

$$C_n = \frac{1}{\sqrt{N}} \left(\sum_{i=1}^n X_i - \frac{n}{N} \sum_{i=1}^N X_i \right) \quad (3)$$

and $\widehat{\Omega}_N$ is the estimator of the asymptotic covariance Ω , where

$$\Omega = \sum_{s=-\infty}^{\infty} \text{Cov}(X_n X_{n-s}). \quad (4)$$

The estimator should satisfy,

$$\widehat{\Omega}_N \xrightarrow{P} \Omega \quad (5)$$

where \xrightarrow{P} denotes convergence in probability.

Several estimators have been proposed in the literature that satisfy (5), including kernel based [13], bootstrap based [14], etc. Considering our requirement for real-time detection (low computational time), a kernel based estimator is more suitable; we employ the Bartlett estimator,

$$\widehat{\Omega}_N = \widehat{\Sigma}_0 + \sum_{l=1}^L k_{BT} \left(\frac{l}{L+1} \right) \left(\widehat{\Sigma}_l + \widehat{\Sigma}_l^T \right) \quad (6)$$

which satisfies (5). The function $k_{BT}(\cdot)$ corresponds to the Bartlett weight,

$$k_{BT}(x) = \begin{cases} 1 - |x|, & \text{for } |x| \leq 1 \\ 0, & \text{otherwise} \end{cases}, \quad (7)$$

while $\widehat{\Sigma}_l$ denotes the empirical auto-covariance matrix for lag l (we chose $L = \log_{10}(N)$ as in [13]),

$$\widehat{\Sigma}_l = \frac{1}{N} \sum_{n=l+1}^N (X_n - \bar{X})(X_{n-l} - \bar{X})^T. \quad (8)$$

The asymptotic covariance is involved in the test statistic to incorporate the dependence structure of the r.v. into the statistical analysis, through the integration of the second order properties. This approach is suitable for the targeted context, since we avoid a restrictive assumption for the dependence structure of the observations.

Going back to the basic question of rejecting or not H_0 , we need to obtain critical values for the test statistic. We approach this issue by considering the asymptotic distribution of the test statistic under H_0 ,

$$M \xrightarrow{D} \sup_{0 \leq t \leq 1} \sum_{j=1}^d B_j^2(t) \quad (N \rightarrow \infty), \quad (9)$$

where \xrightarrow{D} denotes convergence in distribution and $(B_j(t) : t \in [0, 1]), 1 \leq j \leq d$ are independent standard Brownian bridges $B(t) = W(t) - tW(1)$, and $W(t)$ denotes the standard Brownian motion with mean 0 and variance t .

The critical values for several significance levels α can be computed using Monte Carlo simulations that approximate the paths of the Brownian bridge on a fine grid. The last step is to estimate the unknown CP, under H_1 , given by:

$$\widehat{c}_p = \frac{1}{N} \operatorname{argmax}_{1 \leq n \leq N} M. \quad (10)$$

2) *Extended Off-line Approach:* The above hypothesis test answers the problem of the existence of at most one CP and does not ensure that the sample remains statistically constant or stationary in either direction of the detection. In particular, by construction (see equation (2)), the off-line test statistic

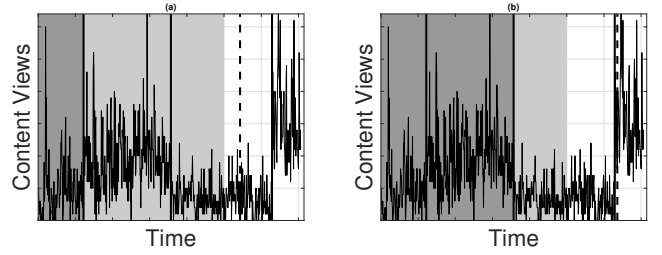


Fig. 1: Importance of training method in the on-line detection (dashed line): a) Basic off-line approach (inaccurate detection). b) Extended off-line approach (accurate detection). Training and excluded samples are denoted with light and dark gray, respectively.

detects the CP with the highest magnitude. Therefore, we need to rephrase the hypothesis test regarding H_1 , as follows:

$$H_1 : \mu_1 = \dots = \mu_{k_1} \neq \mu_{k_1+1} = \dots = \mu_{k_2} \neq \dots \\ \dots \neq \mu_{k_{\tau-1}+1} = \dots = \mu_{k_{\tau}} \neq \mu_{k_{\tau}+1} = \dots = \mu_N.$$

To identify multiple CPs, we developed a heuristic algorithm that combines two well-known methods from the class of binary segmentation (BS): i) the standard BS [15] and ii) the iterative cumulative sum of squares (ICSS) [16] algorithms. These algorithms are computationally efficient ($O(N \log N)$ and $O(N)$ complexities for BS and ICSS, respectively) and easy to implement.

The standard BS algorithm, relies on the general concept of binary segmentation. At first, it searches for a single CP in the dataset. In case of no change, the procedure stops and H_0 is accepted. Otherwise, the detected CP divides the dataset into two sub-datasets and the search is iterated, until no more CPs are detected. An issue with the standard BS is that it overestimates the number of CPs [17].

On the other hand, the ICSS algorithm searches for a single CP in the whole dataset. If a CP is detected, the algorithm searches for a CP in the interval below and above this change, respectively. If H_0 is rejected, the procedure searches for a CP in a new interval with borders the CP (or CPs) previously detected. The algorithm works iteratively until the interval becomes sufficiently small. In order to eliminate the overestimation of CPs, the algorithm searches the potential CPs in pairs and checks if H_0 is still rejected between them, otherwise it eliminates any CP in between.

We merged the two algorithms. In essence, we augmented the standard BS algorithm with the CP validation step of the ICSS. The proposed modified off-line algorithm, including its segmentation process, applies the previously described modified BS procedure in the test statistic M , given in (2).

In Fig. 1 we illustrate the need to involve the modified off-line CP procedure, for an accurate on-line detection. In Fig. 1(a) the standard off-line detector identified a single CP, whereas in reality multiple CPs can be seen to exist. In Fig. 1(b) all CPs have been identified with the modified off-line detector, leading to an accurate detection.

B. On-line Phase

In the following, we describe the on-line scheme that includes: (i) two alternative CUSUM-type approaches for the detection of a change in the mean; and (ii) a moving average convergence divergence (MACD) indicator to estimate the direction of a change.

1) *On-line analysis*: We rewrite equation (1) in the form,

$$X_n = \begin{cases} \mu + Y_n, & n = 1, \dots, m + k^* - 1 \\ \mu + Y_n + I, & n = m + k^*, \dots \end{cases} \quad (11)$$

where $\mu, M \in \mathbb{R}$, represent the mean parameters before and after the unknown time of possible change $k^* \in \mathbb{N}^*$ respectively. The term m denotes the length of the training period, i.e., an interval of length m , during which the mean remains unchanged,

$$\mu_1 = \dots = \mu_m. \quad (12)$$

To satisfy this assumption, the modified off-line CP test previously presented is run and a suitable m is identified. With m determined, the on-line procedure can be used to check whether the assumption (12) holds as new data become available. The importance of a well defined training sample is shown in Fig. 1. In the form of a statistical hypothesis test, the on-line problem is posed as,

$$\begin{aligned} H_0 : I &= 0 \\ H_1 : I &\neq 0 \end{aligned} \quad (13)$$

The on-line sequential analysis belongs to the category of stopping time stochastic processes. In general, a chosen detector $TS(m, k)$ and a given threshold $F(m, k)$ define the stopping time:

$$\tau(m) = \begin{cases} \min\{k \in \mathbb{N} : |TS(m, k)| \geq F(m, k)\} \\ \infty, \text{ otherwise} \end{cases} \quad (14)$$

with the properties,

$$\lim_{m \rightarrow \infty} P\{\tau_m < \infty | H_0\} = \alpha,$$

ensuring that the probability of false alarm is asymptotically bounded by $\alpha \in (0, 1)$, and,

$$\lim_{m \rightarrow \infty} P\{\tau_m < \infty | H_1\} = 1,$$

ensuring that under H_1 the asymptotic power is unity. The threshold $F(m, k)$ is given by,

$$F(m, k) = c_a g_\gamma(m, k). \quad (15)$$

The critical value c_a is determined from the asymptotic distribution of the detector under H_0 and the asymptotic behavior achieved by letting $m \rightarrow \infty$, while the weight function,

$$g_\gamma(m, k) = \sqrt{m} \left(1 + \frac{k}{m}\right) \left(\frac{k}{k+m}\right)^\gamma \quad (16)$$

depends on the sensitivity parameter $\gamma \in [0, 1/2)$.

We apply here two different types of statistics tests, the standard CUSUM [18] and the ratio type CUSUM [19]. The standard CUSUM detector is expressed as:

$$\Gamma(m, k) = \frac{1}{\hat{\omega}_m} \left(\sum_{i=m+1}^{m+k} X_i - \frac{k}{m} \sum_{i=1}^m X_i \right) \quad (17)$$

where $\hat{\omega}_m$ denotes the asymptotic variance, defined in eq. 4, that captures the serial dependence between observations.

The corresponding threshold is $F^\Gamma(m, k) = c_a^\Gamma g_\gamma(m, k)$ and the critical value is defined as:

$$\begin{aligned} \lim_{m \rightarrow \infty} P\{\tau_m < \infty\} &= \lim_{m \rightarrow \infty} P\left\{ \frac{1}{\hat{\omega}_m} \sup_{1 \leq k \leq \infty} \frac{|\Gamma(m, k)|}{g_\gamma(m, k)} > c_a^\Gamma \right\} \\ &= P\left\{ \sup_{t \in [0, 1]} \frac{W(t)}{t^\gamma} > c_a^\Gamma \right\} = a. \end{aligned} \quad (18)$$

Unlike standard CUSUM tests, ratio type statistics do not deal with the long run variance estimator. The precise form of the chosen detector is,

$$\begin{aligned} TR(m, k) &= \\ &= \frac{k^2}{m} D^T|_{m+1}^{m+k} \left(\frac{1}{m^2} \sum_{j=1}^m j^2 \left(D|_1^j D^T|_1^j \right) \right)^{-1} D|_{m+1}^{m+k}, \end{aligned} \quad (19)$$

Threshold function is given by $F^{TR}(m, k) = c_a^{TR} g_\gamma^2(m, k)$, then similar to the standard CUSUM, critical value is given by,

$$\begin{aligned} \lim_{m \rightarrow \infty} P\{\tau_m < \infty\} &= \lim_{m \rightarrow \infty} P\left\{ \sup_{1 \leq k \leq \infty} \frac{|TR(m, k)|}{g_\gamma^2(m, k)} > c_a^{TR} \right\} \\ &= P\left\{ \sup_{t \in [0, \infty)} \Delta_\gamma(t) > c_a^{TR} \right\} = a, \end{aligned} \quad (20)$$

where,

$$\begin{aligned} D|_b^d &= \left(\frac{1}{d-b+1} \sum_{i=b}^d X_i - \frac{1}{m} \sum_{i=1}^m X_i \right), \quad b < d \in \mathbb{N}, \\ \Delta_\gamma(t) &= \frac{1}{\eta_\gamma^2(t)} B^T(1+t) \left(\int_0^1 B(r) B^T(r) dr \right)^{-1} B(1+t), \\ \eta_\gamma^2(t) &= (1+t) \left(\frac{t}{1+t} \right)^\gamma, \end{aligned}$$

and $B(t)$ is a standard Brownian bridge, $t \in [0, \infty)$.

2) *Trend Indicator*: To estimate the direction of change it is not sufficient to look at the immediate neighborhood of a detected CP due to the continuous variability of the time series. We therefore have to estimate the direction of change by incorporating more elaborate filters. In the proposed on-line algorithm, we estimate the direction of detected changes by applying the MACD indicator. MACD is based on an exponential moving average (EMA) filter, of the form,

$$EMA_p(n) = \frac{2}{p+1} X_n + \frac{p-1}{p+1} EMA_p(n-1), \quad (21)$$

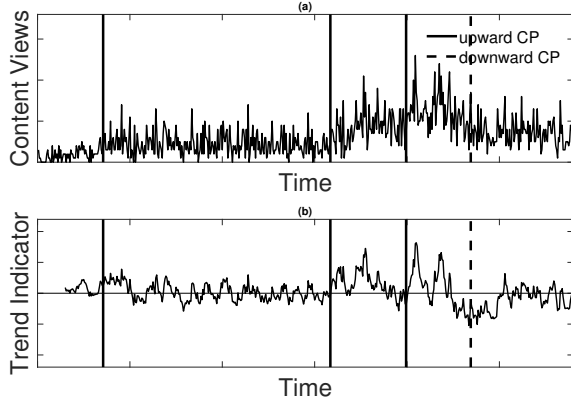


Fig. 2: Off-line CP detection with direction identification. a) Time-series view. b) Equivalent MACD TI view.

with p denoting the lag parameter. The MACD series can be derived from the subtraction from a short p_2 lag EMA (sensitive filter) of a longer p_3 lag EMA (blunt filter), as described below:

$$MACD(n) = EMA_{p_2} - EMA_{p_3}. \quad (22)$$

The trend indicator TI is then obtained by the subtraction of a short p_1 lag EMA filter of a MACD series from the raw MACD series, as described below

$$TI(n) = MACD(n) - EMA_{p_1}(MACD(n)), \quad p_1 < p_2 < p_3. \quad (23)$$

In the evaluation of TI three exponential filters are involved. In essence, TI is an estimation of the second derivative of the change, considering that the subtraction of a filtered variable from the variable generates an estimate of its time derivative. In contrast to other works [20], we only adopt TI to characterize the direction from the specific value of TI at the estimated time of change. We announce an upward change if $TI > 0$, otherwise a downward change. Fig. 2 depicts the operation of TI combined with the CP detector.

Finally, we propose a modification of the trend indicator TI in (23), converting it from a point estimator to an interval estimator; instead of calculating TI at the CP \hat{cp}_i , we propose to evaluate TI at a time interval $(\hat{cp}_i, \hat{cp}_i + h)$, where h is a threshold parameter:

$$TI(\hat{cp}_i, h) = \sum_{k=\hat{cp}_i}^{\hat{cp}_i+h} [MACD(X_k) - EMA_{p_1}(MACD(X_k))]. \quad (24)$$

The proposed modification of the TI estimator improves its accuracy; the calculation of the sum of a multitude of first observations, after a CP, can smooth out a potential false one-point estimation, especially for very small output values (close to zero) of the point estimator.

3) *Overall Algorithm:* Below we describe the pseudo-code of the overall proposed algorithm, comprising 5 steps. We show the algorithm for the content popularity detection and

the interaction between the training phase and the on-line phase. In the proposed load balancing approach we assume that: (i) a new content cache is being deployed, whenever there is an increase in the content popularity; and (ii) an existing content cache is being removed, in the case of a decrease. We outline the integrated algorithm below:

- **Step 1:** From a starting time instance m_s , define a finite monitoring window $k > 0$,
- **Step 2:** Apply the extended off-line algorithm for the whole historical period $h = \{1, m_s\}$,
 - if no changes are detected, the training sample of the sequential procedure becomes $m = h$,
 - else the training sample becomes $m = \{cp_{last}, m_s\}$, where cp_{last} is the last detected off-line CP.
- **Step 3:** Apply the on-line procedure $TS(m, k)$ on the interval $\{m_s, m_s + k\}$
 - if an on-line CP, \hat{cp}^* , is detected the on-line procedure stops,
 - else if no change has occurred set $\hat{cp}^* = 0$ and the monitoring terminates, i.e., proceed to **Step 5**.
- **Step 4:** If $\hat{cp}^* \neq 0$, define $k_{cp} = \hat{cp}^*$ as a CP and apply the trend indicator at the time of change $TI(k_{cp})$,
 - if $TI(k_{cp}) > 0$ then deploy a new content cache (i.e., upward change),
 - else, remove a content cache (i.e., downward change).
- **Step 5:** Set a new starting point for the monitoring period,
 - if $k_{cp} > 0$, set $m_s = k_{cp} + d$, where d is a constant value defining a period assuming no change,
 - else, set $m_s = m_h$.

IV. RESULTS AND PERFORMANCE EVALUATION

In this section, we validate the proposed methods using a dataset of real daily measurements (i.e., 880 youtube video views for 1000 days) extracted from youtube using the tool [21]. In subsection IV-A we conduct extensive experiments, applying the extended off-line algorithm presented in section III-A, to analyze the dataset behavior in terms of content popularity changes and evaluate the off-line algorithm's efficiency to reveal the direction of changes. Studying the detected CP characteristics also serves to adjust the arbitrary parameters of the main algorithm, since the off-line analysis acts as a training phase for the on-line scheme. Finally, we evaluate the proposed on-line method in subsection IV-B.

A. Evaluation of the Extended Off-line Algorithm

We test the hypothesis of no change in the mean structure on our dataset. This results in a high rejection rate, approximately 70%, for a significance level of $\alpha = 0.05$. We also provide the number of content popularity changes, by applying the extended off-line algorithm. The corresponding results are illustrated in Fig. 3 and indicate a sufficiently high number of content popularity anomalies (i.e., mean changes). Hence,

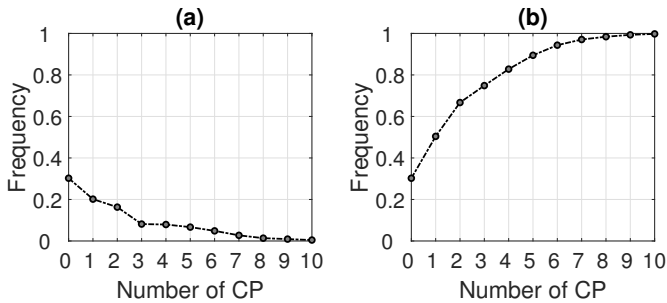


Fig. 3: Estimated a) frequency and b) cumulative frequency of the number of CPs per time-series.

TABLE I: Percentage of Successful *TI* Identification

h	0	3	5	7	10
Video Set 1	0.69	0.91	0.95	0.97	0.98
Video Set 2	0.90	0.99	0.99	0.99	0.99

a CP analysis is indeed a suitable tool for content popularity detection.

To evaluate the performance of the proposed *TI*, we need a baseline independent assessment of direction of change. In the following we assume that: i) a real increase in content visit exists if $mean([X(\hat{c}p_{i-1}), X(\hat{c}p_i)]) < mean(X(\hat{c}p_i) : X(\hat{c}p_{i+1}))$, or that ii) a real decrease in the number of visits exists if $mean(X(\hat{c}p_{i-1}) : X(\hat{c}p_i)) > mean(X(\hat{c}p_i) : X(\hat{c}p_{i+1}))$. We test the modified *TI* on two sets of videos. The first set comprises the whole dataset, while the second set comprises only the videos with a considerable average number of visits (< 10), i.e., $mean(X(1) : X(1000)) > 10$.

The percentage of the successful *TI* identifications are tabulated in Table I for five h thresholds, namely $h = 0, 3, 5, 7$ and 10. Commenting on the results for Video Set 1, the proposed modified *TI* works well, except for $h = 0$, providing at least 90% correct direction identifications. As expected, as the time space increases the procedure works better. More specific, an $h \geq 5$ parameter choice yields a success rate of 95%, while if a more agile estimation is needed then an $h \geq 3$ still maintains a 91% accuracy. Considering the interim time between consecutive changes, analyzed below, we deduce that an $h < 8$ despite the delayed detection remains applicable. Regarding Video Set 2, we see that the results are highly improved, indicating that the procedure works even better for the most popular video. In practice, this represents the more interesting scenario as it will have the greater impact in terms of the applied load balancing mechanism.

Furthermore, in Fig. 4, the time instances of upward and downward changes are shown in the form of a boxplot. It is intuitive that upward changes occurs earlier in time than the downward changes. Moreover, Fig. 4 demonstrates that the multitude of upward changes is greater than the respective of downward changes, indicating that decreases in popularity are sharper than increases. In particular, we estimated that out of the total number of changes, 67% are upward.

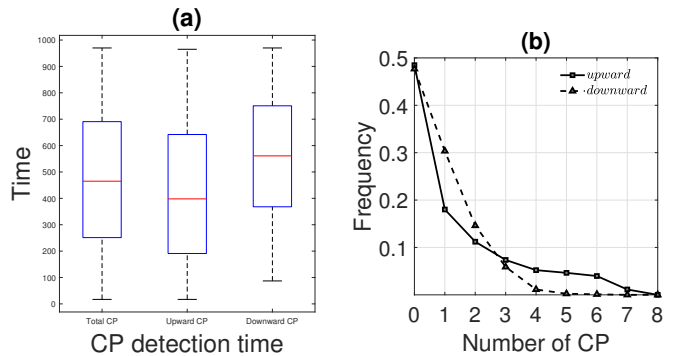


Fig. 4: Frequency values of the number of upward and downward CP, per time series.

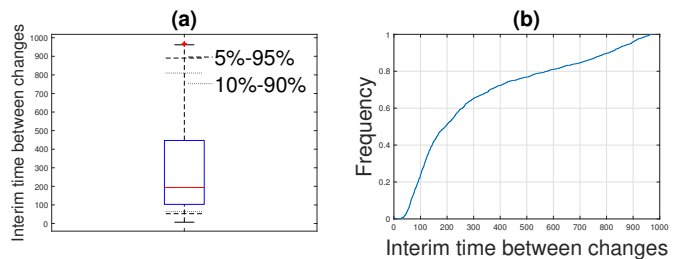


Fig. 5: a) Boxplot including the interval (5% – 95%) (dashed line) and (10% – 90%) interval (dotted line), b) Cumulative frequency for the interim time of consecutive CPs.

Finally, we analyze the interim time between consecutive CPs. Fig. 5 illustrates the existence of a sufficiently large gap between consecutive potential changes. 90% of the intervals corresponding to consecutive CPs exceed 70 time instances, ensuring that a sufficiently large training window can be applied. The results depicted in Fig. 5 allow to adjust parameters of the on-line phase, in particular the minimum time interval between consecutive changes, denoted by the parameter h .

B. Evaluation of the On-line Algorithm

We illustrate the on-line algorithm on different time series. We set the beginning of the monitoring period at $m_s = 200$, the on-line parameter $g = 0.25$ and the significance level $a = 0.95$. The corresponding results are depicted in Figs. 6, 7, showing the estimated changes by applying the standard CUSUM procedure and the ratio type CUSUM process, respectively. In both cases, the estimated changes correspond to the real mean changes. In fact, in some cases, the ratio type test seem to overestimate the number of changes (e.g., Fig. 7d). A more exhaustive comparison is needed between the two schemes, for a more comprehensive evaluation.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we developed a novel algorithm for the real-time detection of content popularity changes. Approaching the problem statistically, we efficiently combined off-line and on-line non-parametric CUSUM procedures to avoid restrictive assumptions for content popularity behavior and to reduce

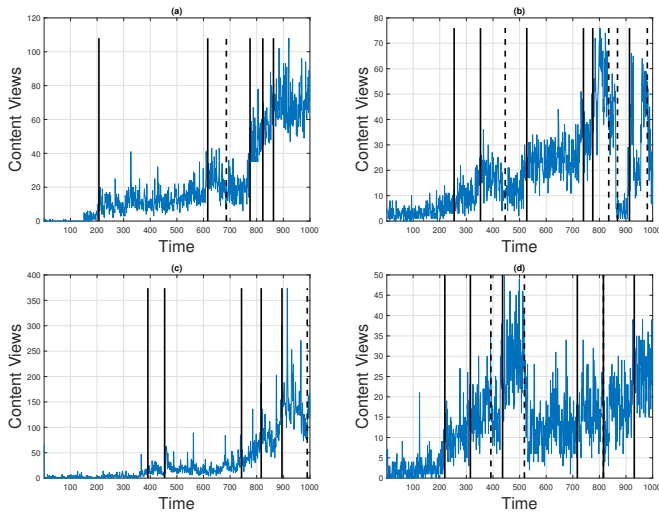


Fig. 6: On-line standard CUSUM algorithm. Solid and dashed lines depict an upward and a downward change, respectively.

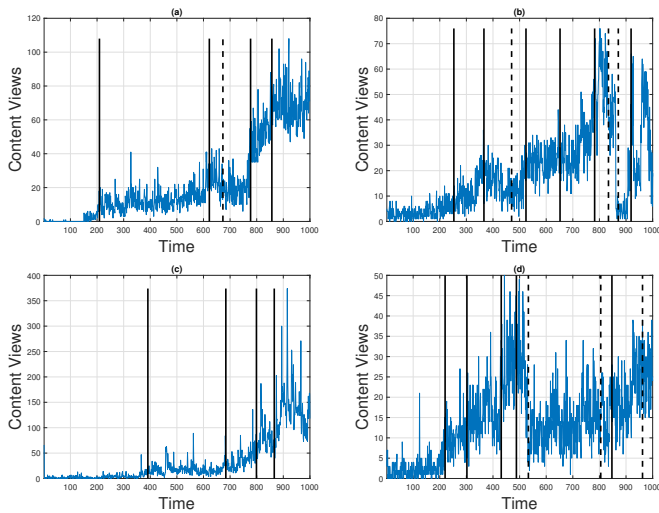


Fig. 7: On-line ratio CUSUM algorithm. Solid and dashed lines depict an upward and a downward change, respectively.

the overall computational cost. We divided the algorithm in two phases. The first phase is an extended retrospective (off-line) procedure with an improved BS algorithm and is used to adjust on-line parameters, based on historical data of the particular video. The second phase integrates a modified trend indicator to the sequential (on-line) procedure, to reveal the direction of a detected change. We provided extensive simulations, using real data, that demonstrate the performance of the first phase of our algorithm. We also provided proof-of-concept results that highlight the efficiency of the overall algorithm. In future work, we will improve the proposed scheme by incorporating multiple dimensions in the existing methodology that could represent other parameters related to user's surfing habits (e.g., likes, comments, etc.), or real-time resource utilization of servers. We will also investigate

the scalability behavior of our solution, theoretically and experimentally, i.e., estimate the number of videos it can analyze in parallel. Our goal is a distributed architecture with multiple content popularity analyzer entities, investigating clusters of videos with minimum overall processing cost.

ACKNOWLEDGMENTS

This work is partially supported by the open call scheme of the H2020 MONROE (grant agreement 644399) project and the H2020 4th EU-BR Collaborative Call NECOS - Novel Enablers for Cloud Slicing (grant agreement no. 777067).

REFERENCES

- [1] CISCO Visual Networking Index, *Forecast Highlights, Cisco Visual Networking Index: Forecast*, pp. 2016-2021, 2017.
- [2] Novel Enablers for Cloud Slicing (NECOS), H2020 Project, EU-Brazil Joint Call EUB-01-2017, <http://www.h2020-necos.eu/>
- [3] P. Valsamas, S. Skaperas and L. Mamatras. "Elastic Content Distribution Based on Unikernels and CP Analysis," *IEEE European Wireless Conf.*, 2-4 May, 2018, Catania, Italy.
- [4] A. Tatar *et al.*, "A survey on predicting the popularity of web content" *J. Internet Services Appl.*, vol. 5, no 1, 8, 2014.
- [5] G. Szabo and B.A. Huberman, "Predicting the popularity of online content", in *Proc. Commun ACM 2010*, vol. 53, no 8, pp. 80-88.
- [6] H. Pinto *et al.*, "Using early view patterns to predict the popularity of youtube videos" in *Proc. 6th ACM Int. Conf. Web Search Data Mining, WSDM 13*, ACM, Rome, Italy.
- [7] G. Gursun *et al.*, "Describing and forecasting video access patterns", in *INFOCOM*, 2011, Shanghai, China.
- [8] M. Ahmed *et al.*, "A peek into the future: predicting the evolution of popularity in user generated content", in *Proc. 6th ACM Int. Conf. Web Search Data Mining, WSDM 13*, ACM, Rome, Italy.
- [9] X. Zhou, and C.Z. Xu, "Optimal video replication and placement on a cluster of video-on-demand servers", in *Proc. IEEE Int. Conf. Parallel Process.*, pp. 547555, 2002.
- [10] W. Tang *et al.*, "Modeling and generating realistic streaming media server workloads," *Comput. Netw.*, vol. 51, no 1, pp. 336-356, 2007.
- [11] Y. Jiang *et al.*, "A novel caching policy with content popularity prediction and user preference learning in Fog-RAN," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2017, pp. 1-6, 2017.
- [12] A. Aue, and L. Horvth, "Structural breaks in time series," *J. Time Series Analysis*, vol. 34, no. 1, pp. 1-16, 2013.
- [13] D. W. K. Andrews, "Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation," *Econometrica*, Vol. 59, No. 3, pp. 817-858, 1991.
- [14] D. Wied, "A nonparametric test for a constant correlation matrix," *Econometric Reviews*, vol. 36, no. 10, pp. 1157-1172, 2017.
- [15] L. J. Vostrikova, "Detecting 'disorder' in multidimensional random processes," *Doklady Akademii Nauk SSSR*, vol. 259, no. 2, pp. 270-274, 1981.
- [16] C. Inclan and G.C. Tiao, "Use of cumulative sums of squares for retrospective detection of changes of variance", *J. American Stat. Assoc.*, vol. 89, no. 427, pp. 913-923, 1994.
- [17] A. Badagin *et al.*, "Time series segmentation procedures to detect, locate and estimate CPs," *Empirical Economic and Financial Research*, Springer, pp. 45-59, 2015.
- [18] S. Fremdt, "Asymptotic distribution of the delay time in page's sequential procedure," *J. Stat. Planning Inference*, vol. 145, pp. 74-91, 2014.
- [19] Y. Hoga, "Monitoring multivariate time series," *J. Multivariate Analysis*, vol. 155, pp. 105-121, 2017.
- [20] H. Kai *et al.*, "Network anomaly detection based on statistical approach and time series analysis," in *Proc. IEEE Int. Conf. Advanced Information Netw. Appl. Workshops WAINA'09*, 2009, pp. 205-211, 2009.
- [21] M. Zeni *et al.*, "YOUStatAnalyzer: a tool for analysing the dynamics of YouTube content popularity," in *Proc. 7th Int. Conf. Performance Evaluation Methodologies Tools, ICST*, pp. 286-289, 2013.