On Moving Averages, Histograms and Time-Dependent Rates for Online Measurement

Michael Menth and Frederik Hauser
Department of Computer Science
University of Tuebingen, Germany
{menth, frederik.hauser}@uni-tuebingen.de

ABSTRACT

Moving averages (MAs) are often used in adaptive systems to monitor the state during operation. Their output is used as input for control purposes. There are multiple methods with different ability, complexity, and parameters. We propose a framework for the definition of MAs and develop performance criteria, e.g., the concept of memory, that allow to parameterize different methods in a comparable way. Moreover, we identify deficiencies of frequently used methods and propose corrections. We extend MAs to moving histograms which facilitate the approximation of time-dependent quantiles. We further extend the framework to rate measurement, discuss various approaches, and propose a novel method which reveals excellent properties. The proposed concepts help to visualize time-dependent data and to simplify design, parametrization, and evaluation of technical control systems.

1. INTRODUCTION

Moving averages (MAs), moving histograms, and time-dependent rates calculate time-dependent statistics from time series while giving more importance to recent than to old samples. The exponential MA (EMA) is a simple example:

$$A_i = a \cdot A_{i-1} + (1-a) \cdot X_i. \quad (1)$$

X_i is the size of an observed sample at time i and A_i is the time-dependent average at that time. The smoothing parameter a is often set to $a = 0.9$, but there is only little insight in literature about appropriate configuration.

On the one hand, the above mentioned methods are helpful to track and visualize the behavior of technical systems over time. On the other hand, they are used for control purposes by adaptive systems to observe their state and react appropriately. A prominent example is TCP [1] which estimates the current roundtrip time (RTT) by exponential smoothing of individual RTT samples. As another example, we currently develop an automatic bypass for a rate-constraint firewall using software-defined networking (SDN). The firewall is attached to a border switch which steers all in- and outgoing traffic through the firewall. Traffic leaving the firewall is monitored by exporting every n^{th} packet to the SDN controller using sFlow. The controller calculates a time-dependent rate over a relatively short time scale to detect congestion when a rate threshold is exceeded. Then, the controller bypasses sampled flows around the firewall through installation of appropriate flow rules on the switch. As the switch can support only a limited number of rules, the installation rate may be high if congestion occurs rather seldom, and should be low otherwise. To that end, we track congestion using a MA over a relatively long time scale and use its value for computation of a suitable offloading rate.

In this paper, we consider multiple MA methods with different parameters. We define some metrics for MAs that relate to time scale. The memory is most important and helps to set the parameters of the different methods such that their output behaves similarly. MAs are mostly applied under the assumption that samples are equidistantly spaced in time (evenly spaced time series). However, in practice unevenly spaced time series may also occur. Therefore, we adapt well-known MA methods such that the time between samples has influence on calculated average values. We show that EMA has a strong bias towards the first observed sample X_0 and propose a method for an unbiased EMA (UEMA). We demonstrate that EMA's existing extension for unevenly spaced time series (TEMA) is even persistently biased and suggest an unbiased TEMA (UTEMA). A challenge for the application of MAs is the tradeoff between accuracy and timeliness: MAs require a large memory to produce accurate estimates for mean values which is desired under the assumption of a stationary process. However, they need a short memory to quickly reveal changed behavior of a non-stationary process. We illustrate this tradeoff and give recommendations for appropriate parametrization. We propose moving histograms (MHs) as a straightforward extension of MAs including an efficient implementation. They allow approximation of time-dependent quantiles. Time-dependent rates should reflect the recent intensity of a sample process. We define time-dependent rate measurement (TDRM) as an extension of MAs. The time scale of TDRM can be controlled by the memory of the underlying MA. We review and compare existing TDRM methods and suggest TDRM-UTEMA as a novel method which exhibits desired properties.

The authors acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG) under grant ME2727/2-1.

ICPE'17, April 22-26, 2017, L'Aquila, Italy
© 2017 ACM. ISBN 978-1-4503-4404-3/17/04…$15.00
DOI: http://dx.doi.org/10.1145/3030207.3030212

We believe that this rather elementary work contributes to a more informed usage of MAs, MHs, and TDRM. It focuses on online measurement, i.e., only the past of the process is known at measurement time. In contrast, offline measurement can consider a complete time series as input and smooth values or determine rates by leveraging both past and future samples around an observation point. While this work considers time series in the time domain, there is a large body of related work for time series analysis in the frequency domain [2].

The paper is structured as follows. Section 2 shows that MAs are used in various fields with different applications. Section 3 provides a novel framework to define MAs, suggests novel metrics to characterize MA properties, in particular the memory, reviews well-known MA approaches, and proposes new methods. Section 4 studies the impact of memory on the accuracy and timeliness of UEMA. MAs are extended towards MHs in Section 5. An extension for TDRM is proposed in Section 6: existing methods and a novel one are presented and compared. Section 7 concludes this work.

2. RELATED WORK

Though exponential smoothing is a standard technique for scientific work, it is hard to track back its roots in scientific literature. Books about fundamental statistics like [3] present only unweighted (simple) and weighted moving averages. However, we did not find an overview of general MA concepts and their properties. MAs are known under a broad range of different terms, e.g., they are also referred to as smoothing or filtering methods. Eckner describes simple and exponential moving averages as well as rolling operators [4] for the purpose of smoothing. The work in [5] presents similar concepts for the specific requirements of unevenly spaced time series. Autoregressive MAs and variants are discussed to model stochastic processes for the purpose of forecasting which is different from our application which is the calculation of an average value for the current observation point.

MAs are often applied in networking. Three different modifications of EMA (low pass, gradient adaptive, and retrospective) for bandwidth estimation are presented in [6]. Conga [7] is a distributed congestion-aware load balancing mechanism for datacenters. It leverages a discounting rate estimator which is similar to EMA. CSFQ [8] aims at providing fair bandwidth allocation. Exponential averaging with variable weights is used to estimate flow arrival rates. The active queue management PIE [9] is designed to control latency and jitter in the Internet. Its departure rate estimation uses exponential smoothing. TCP's smoothed roundtrip time (SRTT) mechanism computes the retransmission timeout in data communication with exponential averaging [1]. In [10] we presented a rate measurement method based on exponential smoothing. It was leveraged by the authors in [11] for time-decaying Bloom filters. Furthermore, we introduced the concept of moving histograms in [12] to calculate time-dependent quantiles.

MAs also have application in other areas. A simple window-based moving average is used in [13] for detection of the end of the transient phase of a stochastic process. To that end, a symmetric moving window is applied as low-pass filter to a time series, extracting its long-term trend. Exponential smoothing is widely used in the context of operations research and financial analyses for trend forecasting (e.g. [14]). The Holt-Winters forecasting procedure [15] uses three degrees of smoothing to extract level, trend, and seasonal components in time series. In quality control, exponential smoothing with optimized weights is used for the generation of control charts which detect exceedance or shortfall of critical boundaries [16, 17].

3. MOVING AVERAGES (MA)

We consider MAs for samples observed with evenly and unevenly spaced time series. For both cases, we propose a general definition of MAs. We introduce several performance metrics to characterize properties of MAs. We consider specific MAs and express their properties depending on their parameters. We point out differences among presented MAs, show that widely used exponential moving averages for evenly and unevenly spaced time series have an initial or even persistent bias, and propose unbiased variants.

3.1 MAs for Evenly Spaced Time Series

Let $(X_i)_{0\leq i<\infty}$ be an evenly spaced time series with samples of size X_i and Δt time between consecutive samples (inter-sample time). We define a MA A_j for observation point j by

$$S_j = \sum_{0\leq i\leq j} g_i(i-j)\cdot X_i \qquad (2)$$

$$N_j = \sum_{0\leq i\leq j} g_i(i-j) \qquad (3)$$

$$A_j = \begin{cases} \frac{S_j}{N_j} & N_j > 0 \\ 0 & \text{otherwise.} \end{cases} \qquad (4)$$

The sample-specific discrete weight functions $g_i(.)$ are characteristic for special types of MAs. They are used to compute a weighted sample sum S_j and weighted sample number N_j. Thereby, the weight functions $g_i(.)$ may reduce the impact of samples X_i on the MA that are distant from the observation point j. For most MA types, the weight $g_i(0)$ for the most recent sample takes the maximum $g_i^{max} = \max_k(g_i(k))$. The function decreases monotonously towards negative and positive arguments whereby only negative arguments are considered for online measurement. The fraction of the weighted sample sum and the weighted sample number yields the MA A_j if the number N_j is positive. Otherwise, we define the MA to be zero.

3.1.1 Metrics

If the weight functions $g_i(.)$ for samples X_i do not differ, their subscript as well as those of the following metrics may be omitted.

The contribution C_i quantifies how much a sample X_i contributes to all average values A_j. It can be calculated by

$$C_i = \sum_{-\infty<k\leq 0} g_i(k)\cdot\Delta t. \qquad (5)$$

If contributions C_i differ, the MA has a bias to towards samples with larger contributions.

The memory M_i quantifies the average duration over which a sample X_i contributes to average values A_j. It considers fractional contributions relative to the maximum sample weight g_i^{max} and can be derived as

$$M_i = \frac{C_i}{g_i^{max}}. \qquad (6)$$

Essentially, the memory reflects the time scale over which samples are averaged. If weight functions $g_i(.)$ differ among samples, they may – but do not need to – yield different contributions C_i and memory M_i.

The memory depends on the inter-sample time Δt which must be taken into account when applying MAs in practice. A MA may be used to track the transmission delay of packets on a communication line. Transmission of packets with 1500 bytes yields inter-sample times of $\Delta t = 12$ ms and $\Delta t = 0.012$ ms on a 1 Mb/s and 1 Gb/s link, respectively. Irrespective of the specific type of MA, the resulting memory differs by three orders of magnitude if MAs are applied with identical parametrization. If the timely dynamics of the measured averages should be comparable, the parameters of the MAs need to be adapted to the specific inter-sample time Δt so that the MAs exhibit the same memory for all considered processes.

The delay D_i quantifies the average age of all contributions of a sample X_i to all average values A_j. It is computed by

$$D_i = \sum_{-\infty < k \leq 0} (|k| \cdot \Delta t) \cdot \frac{(g_i(k) \cdot \Delta t)}{C_i}. \tag{7}$$

Shifting a MA's weight function by j to $g_i^*(k) = g_i(k+j)$ for $k \leq -j$ and $g_i^*(k) = 0$ for $k > -j$ leads to another MA with the same memory but a larger delay.

3.1.2 Cumulative Mean (CumMean)

CumMean is defined by

$$A_j = \frac{1}{j+1} \cdot \sum_{0 \leq i \leq j} X_i. \tag{8}$$

It fits the Definitions (2) – (4) for homogeneous weights $g(k) = 1$, $-\infty < k \leq 0$. Thus, each sample exhibits a contribution, delay, and memory of $M = D = C = \infty \cdot \Delta t$. Figure 1(a) visualizes the evolution of the CumMean for the evenly spaced time series $(X_0, ..., X_{11}) = (1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0)$. The impact of recent samples diminishes with increasing time index because all past samples equally contribute to CumMean's average. Therefore, CumMean does not focus on the recent evolution of the observed process and cannot provide appropriate feedback for self-adapting systems.

3.1.3 Window Moving Average (WMA)

WMA, also known as simple moving average (SMA) [4], essentially computes the arithmetic mean of the last w samples whereby w is an integer. The weighted sample sum is

$$S_j = \sum_{max(0, j-w+1) \leq i \leq j} X_i \tag{9}$$

and the weighted sample number is $N_j = min(w, j+1)$. This fits the Definitions (2) – (4) for weights

$$g(k) = \begin{cases} 1 & -w < k \leq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

Contribution and memory are $C = M = w \cdot \Delta t$ and delay is $D = \frac{(w-1) \cdot \Delta t}{2} = \frac{M}{2} - \frac{\Delta t}{2}$.

Figure 1(a) visualizes the evolution of WMA-based averages for the above introduced sample series. It yields an average value of $A_{11} = 0$ because it disregards past samples outside its window. Furthermore, it yields $A_5 = 0.75$ and $A_6 = 0.75$ although the observed '1' are younger in A_5 than in A_6. This is due to the fact that all samples within a WMA's window are equally weighted.

3.1.4 Disjoint Windows Moving Average (DWMA)

DWMA partitions a time series into sets of consecutive and disjoint windows $\mathcal{W}_k = \{k \cdot w \leq i < (k+1) \cdot w\}$ with w time indices each. At the end of such a window, the arithmetic mean of the samples falling in this window is taken as current average until the next measurement value is available. This can be denoted by

$$S_j = \begin{cases} 0 & j < w-1 \\ \sum_{i \in \mathcal{W}_{(\lfloor \frac{j+1}{w} \rfloor - 1)}} X_i & \text{otherwise} \end{cases} \tag{11}$$

and $N_j = w$. It fits the Definitions (2) – (4) for sample-specific weights

$$g_i(k) = \begin{cases} 1 & \begin{cases} (i \mod w) - (2 \cdot w - 1) \\ < k \leq \\ (i \mod w) - (w-1) \end{cases} \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Thus, contribution and memory are $C = M = w \cdot \Delta t$. The sample-specific delay is $D_i = \frac{w-1}{2} \cdot \Delta t + ((w-1) - (i \mod w)) \cdot \Delta t$ and the average delay is $\overline{D} = (w-1) \cdot \Delta t = M - \Delta t$.

Figure 1(a) shows the evolution of DWMA. For the first $w - 1$ observation points, there is no actual average value available. Due to its increased delay, DWMA clearly lags behind WMA and it is coarser.

3.1.5 Unbiased Exponential Moving Average (UEMA)

We propose UEMA as a novel algorithm using the weight function

$$g(k) = a^{-k} \tag{13}$$

and the Definitions (2) – (4) for the computation of the average values. The underlying geometric model enables an elegant, recursive calculation:

$$S_j = \begin{cases} X_j & j = 0 \\ a \cdot S_{j-1} + X_j & j > 0 \end{cases} \tag{14}$$

$$N_j = \begin{cases} 1 & j = 0 \\ a \cdot N_{j-1} + 1 & j > 0 \end{cases}. \tag{15}$$

UEMA has a contribution and memory of $C = M = \frac{\Delta t}{1-a}$, and a delay of $D = \frac{a \cdot \Delta t}{1-a}$ for all samples, i.e., $D = a \cdot M$. Figure 1(a) compares the evolution of UEMA with the one of WMA for the same memory $M = 4 \cdot \Delta t$. While WMA disregards samples that lie outside its window and yields $A_{11} = 0$, UEMA respects the full past of the process and yields $A_{11} = 0.21$. WMA also disregards the position of samples within its window. In contrast, UEMA is sensitive to that and yields $A_5 = 0.87$ (observed 0 in WMA's window is old) and $A_6 = 0.62$ (observed 0 in WMA's window is young).

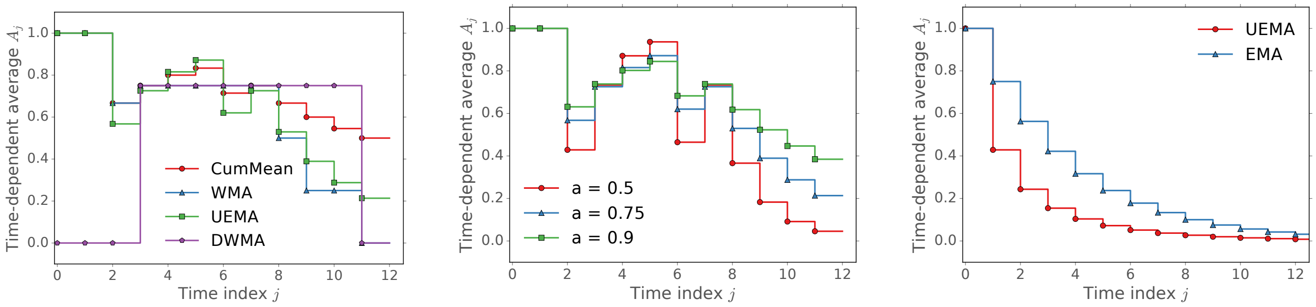
Figure 1(b) illustrates the impact of UEMA's memory on the evolution of the MA. The MA is more inert for larger memory, i.e., for a larger smoothing factor a.

3.1.6 Exponential Moving Average (EMA)

EMA, also known as Exponentially Weighted MA (EWMA) or exponential smoothing, calculates its weighted sum S_j as

$$S_j = \begin{cases} X_0 & j = 0 \\ a \cdot S_{i-1} + (1-a) \cdot X_j & j > 0 \end{cases} \tag{16}$$

and the weighted number of samples is $N_j = 1$. Thus, S_j already computes A_j (see Equation (1)). These formulae fit



(a) CumMean ($M = \infty$), WMA, DWMA (both with $w = 4$, $M = 4 \cdot \Delta t$), and UEMA ($a = \frac{3}{4}$, $M = 4 \cdot \Delta t$) for time series $(X_0, ..., X_{11}) = (1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0)$.

(b) Comparison of UEMA for $a \in \{\frac{1}{2}, \frac{3}{4}, \frac{9}{10}\}$ ($M \in \{2 \cdot \Delta t, 4 \cdot \Delta t, 10 \cdot \Delta t\}$) for time series $(X_0, ..., X_{11}) = (1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0)$.

(c) EMA and UEMA with $a = 0.75$ ($M = 4 \cdot \Delta t$) for time series $(X_0, ..., X_{11}) = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$: EMA has an initial bias.

Figure 1: Timely evolution of MAs for evenly spaced time series.

the Definitions (2) – (4) for sample-specific weights

$$g_i(k) = \begin{cases} a^{-k} & i = 0 \\ (1-a) \cdot a^{-k} & \text{otherwise} \end{cases}. \tag{17}$$

The two different weight functions yield the same memory $M = \frac{\Delta t}{1-a}$ and delay of $D = \frac{a \cdot \Delta t}{1-a}$ which equal those of UEMA. However, the contribution of X_0 is $C_0 = \frac{\Delta t}{1-a}$ while the one of all other samples is $C_i = \Delta t$. This causes a bias towards X_0 in the time series of the resulting MA A_j. To illustrate this effect, Figure 1(c) compares the resulting averages for EMA and UEMA for consecutive samples $(X_0, ..., X_{11}) = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. EMA and UEMA are both configured with $a = 0.75$, i.e., $M = 4 \cdot \Delta t$. The difference between the two curves is the bias induced by EMA's larger contribution of X_0. As the bias vanishes over time, EMA may be used if accuracy for initial values A_j does not matter. This may be advantageous as EMA is slightly simpler than UEMA. While EMA is already applied in many technical systems and research papers, UEMA is a novel method proposed in this work.

3.2 MAs for Unevenly Spaced Time Series

Let $(X_i)_{t_i \in \mathcal{T}, 0 \le i < \infty}$ be an unevenly spaced time series of samples with different size X_i. Figure 2(a) contrasts two examples to an evenly spaced time series.

MAs for unevenly spaced time series respect the time structure of samples such that the impact of a sample on resulting average values diminishes over time instead with progressing time index. We define a MA A_t for observation point t by

$$S_t = \sum_{\{i: t_i \le t\}} g_i(t_i - t) \cdot X_i \tag{18}$$

$$N_t = \sum_{\{i: t_i \le t\}} g_i(t_i - t) \tag{19}$$

$$A_t = \begin{cases} \frac{S_t}{N_t} & N_t > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{20}$$

Definitions (18) – (20) are analogous to Definitions (2) – (4) but account for the fact that sample times are now continuous. Therefore, the sample-specific weight functions $g_i(.)$ are also continuous.

3.2.1 Metrics

The metrics are also analogous to those of MAs for evenly spaced time series. The contribution C_i of sample X_i can be calculated as

$$C_i = \int_{-\infty}^{0} g_i(t)dt \tag{21}$$

and the memory is

$$M_i = \frac{C_i}{g_i^{max}} \tag{22}$$

with $g_i^{max} = \max_{-\infty < t \le 0}(g_i(t))$. The delay is

$$D_i = \frac{\int_{-\infty}^{0} |t| \cdot g(t)dt}{C_i}. \tag{23}$$

3.2.2 Time Window Moving Average (TWMA)

TWMA computes the average of the samples within a recent time window of duration W. Let $\mathcal{W}_t = \{i : t_i \in (t - W; t]\}$ be the index set of samples arriving within that window at time t. Average values for TWMA can be computed by $N_t = |\mathcal{W}_t|$ and

$$S_t = \begin{cases} 0 & |\mathcal{W}_t| = 0 \\ \sum_{i \in \mathcal{W}_t} X_i & \text{otherwise} \end{cases}. \tag{24}$$

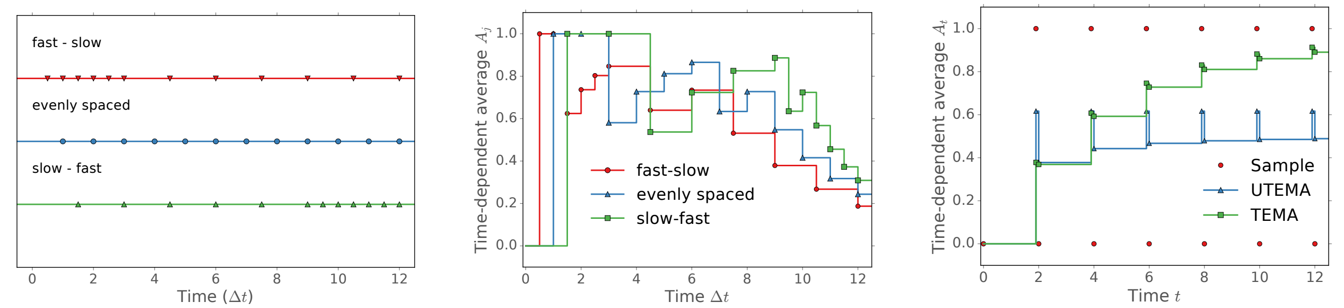
This fits the Definitions (18) – (20) for the weight function

$$g(t) = \begin{cases} 1 & -W < t \le 0 \\ 0 & \text{otherwise} \end{cases}. \tag{25}$$

Contribution and memory are $C = M = W$ and the delay is $D = \frac{W}{2} = \frac{M}{2}$. The resulting MA A_t returns to zero if the last sample is older than W, which may be an undesired property. We omit illustrations of TWMA due to space limitations.

3.2.3 Disjoint Time Windows Moving Average (DTWMA)

DTWMA partitions the time axis into consecutive measurement intervals of duration W, computes the arithmetic mean of samples observed within a measurement interval at its end, and uses this average value until the next average value is available. Let $\mathcal{W}_k = \{i : t_i \in (k \cdot W; (k+1) \cdot W]\}$ be the index set of samples arriving within window number k since measurement start at time $t = 0$. Average values for DTWMA are computed by $N_t = |\mathcal{W}_{(\lceil \frac{t}{W} \rceil - 1)}|$ and



(a) Three considered arrival times: samples first arrive fast and then slowly, vice-versa, and at constant speed.

(b) UTEMA with $\beta = \frac{1}{4 \cdot \Delta t}$ ($M = 4 \cdot \Delta t$) for time series $(1,1,0,1,1,1,0,1,0,0,0,0)$ and the arrival times in Figure 2(a).

(c) TEMA and UTEMA with $\beta = \frac{1}{\Delta t}$ ($M = 1 \cdot \Delta t$): TEMA exhibits an initial and a persistent bias. Sample arrivals and sizes are represented by dots.

Figure 2: Timely evolution of MAs for unevenly spaced time series.

$$S_t = \begin{cases} 0 & t < W \vee |\mathcal{W}_{(\lceil \frac{t}{W} \rceil - 1)}| = 0 \\ \sum_{i \in \mathcal{W}_{(\lceil \frac{t}{W} \rceil - 1)}} X_i & \text{otherwise} \end{cases} . \quad (26)$$

These equations fit Definitions (18) – (20) for sample-specific weight functions

$$g_i(t) = \begin{cases} 1 & \begin{cases} t_i - (\lceil \frac{t_j}{W} \rceil + 1) \cdot W \\ < t \leq \\ t_i - \lceil \frac{t_j}{W} \rceil \cdot W \end{cases} \\ 0 & \text{otherwise} \end{cases} . \quad (27)$$

Thus, contribution and memory are $C_i = M_i = W$. The sample-specific delay is $D_i = (\lceil \frac{t_i}{W} \rceil \cdot W - t_i) + \frac{W}{2}$ and the average delay is $\overline{D} = W$. The MA A_t returns to zero after a measurement interval without any samples. We omit illustrations of DTWMA due to space limitations.

3.2.4 Unbiased Time-Exponential Moving Average (UTEMA)

We propose UTEMA as a novel MA method which uses the exponential weight function

$$g(t) = e^{\beta \cdot t} \quad (28)$$

in Definitions (18) – (20). The underlying exponential model allows for recursive equations:

$$S_t = \begin{cases} 0 & t < t_0 \\ X_0 & t = t_0 \\ e^{-\beta \cdot (t - t_{i-1})} \cdot S_{t_{i-1}} + X_i & t = t_i \\ e^{-\beta \cdot (t - t_i)} \cdot S_{t_i} & t_i < t < t_{i+1} \end{cases} \quad (29)$$

$$N_t = \begin{cases} 0 & t < t_0 \\ 1 & t = t_0 \\ e^{-\beta \cdot (t - t_{i-1})} \cdot N_{t_{i-1}} + 1 & t = t_i \\ e^{-\beta \cdot (t - t_i)} \cdot N_{t_i} & t_i < t < t_{i+1} \end{cases} . \quad (30)$$

UTEMA has a contribution, memory, and delay of $C = M = D = \frac{1}{\beta}$.

Figure 2(b) shows that UTEMA respects the time structure of the sample processes given in Figure 2(a). If samples first arrive fast and then slowly, UTEMA yields a lower average $A_{t=12 \cdot \Delta t}$ at the end of the observation interval than for the evenly spaced time series because observed '1' are older. Likewise, if samples first arrive slowly and then fast, UTEMA leads to a larger $A_{t=12 \cdot \Delta t}$ than for the evenly spaced time series because observed '1' are younger. When UEMA is applied for one of the unevenly spaced time series, it yields UTEMA's average values for the evenly spaced time series.

UEMA can provide exactly the same average values for an evenly spaced time series as UTEMA if its smoothing factor is set such that it yields the same weights for integral multiples of Δt, i.e., $a = e^{-\beta \cdot \Delta t}$. Figure 3 compares such weight functions of UTEMA and UEMA with a smoothing rate of $\beta = \frac{0.25}{\Delta t}$ and smoothing factor of $a = 0.7788$, respectively. The memory for UTEMA is $M = 4 \cdot \Delta t$ while the one for UEMA is $M = 4.52 \cdot \Delta t$. The discrepancy is due to the fact that UTEMA's weight function $g_{UTEMA}(t)$ interpolates only the lower corners of UEMA's weights $g_{UEMA}(k)$ and, therefore, leads to a lower integral value. This difference converges to $\frac{\Delta t}{2}$ for large memory M.

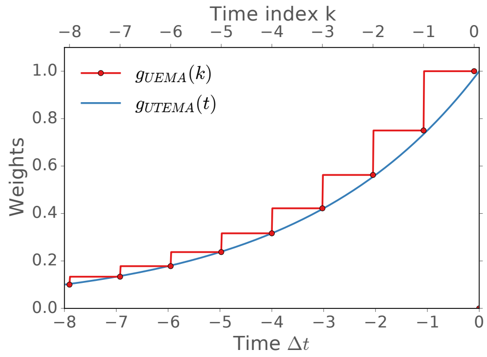


Figure 3: Weight functions for UEMA and UTEMA configured such that they reveal equal weights for integral multiples of Δt ($a = 0.7788$ and $\beta = \frac{0.25}{\Delta t}$, respectively). UTEMA yields a lower memory because its weight function $g_{UTEMA}(t)$ interpolates only the lower corners of UEMA's step function $g_{UEMA}(k)$.

The difference between UTEMA's average curves for evenly and unevenly spaced time series quantifies the error caused by UEMA. Its practical significance depends on the time scale of interest because the difference decreases for increasing memory M. If the time structure of the observed process is essential, UTEMA is a good alternative to UEMA and also to TWMA which does not respect the time struc-

Table 1: Considered MAs including properties.

Method	Param.	Contribution C	Memory M	Delay D	Delay D dep. on M	Comments
MAs for evenly spaced time series						
CumMean	Δt	∞	∞	∞	∞	Does not discount old samples.
WMA	$w, \Delta t$	$w \cdot \Delta t$	$w \cdot \Delta t$	$\frac{(w-1)\cdot\Delta t}{2}$	$\frac{M}{2} - \frac{\Delta t}{2}$	Does not account for samples outside window and sample position within window.
DWMA	$w, \Delta t$	$w \cdot \Delta t$	$w \cdot \Delta t$	$(w-1) \cdot \Delta t$ (avg.)	$M - \Delta t$ (avg.)	Like WMA, additional delay
UEMA	$a, \Delta t$	$\frac{\Delta t}{1-a}$	$\frac{\Delta t}{1-a}$	$\frac{a\cdot\Delta t}{1-a}$	$a \cdot M$	Method proposed in this work
EMA	$a, \Delta t$	$\frac{\Delta t}{1-a}$, Δt	$\frac{\Delta t}{1-a}$	$\frac{a\cdot\Delta t}{1-a}$	$a \cdot M$	Bias towards X_0
MAs for unevenly spaced time series						
TWMA	W	W	W	$\frac{W}{2}$	$\frac{M}{2}$	May return to zero (invalid value).
DTWMA	W	W	W	W (avg.)	M (avg.)	Like TWMA, additional delay
UTEMA	β	$\frac{1}{\beta}$	$\frac{1}{\beta}$	$\frac{1}{\beta}$	M	Method proposed in this work
TEMA	β	$\frac{1}{\beta}$ for X_0, $\frac{1-e^{-\beta\cdot(t_i-t_{i-1})}}{\beta}$ for X_i	$\frac{1}{\beta}$	$\frac{1}{\beta}$	M	Bias towards X_0 and other samples

ture within its measurement window and may return to zero in the absence of sufficiently young samples.

For UTEMA, the half-life time may be used as another metric. It can be computed by $H = \frac{\ln(2)}{\beta}$, but it is not applicable to most other MA variants.

3.2.5 *Time-Exponential Moving Average (TEMA)*

TEMA is sometimes used as adaptation of EMA to unevenly spaced time series [4]. Its sample sum is computed by

$$S_t = \begin{cases} 0 & t < t_0 \\ X_0 & t = t_0 \\ e^{-\beta\cdot(t_i - t_{i-1})} \cdot S_{t_{i-1}} + & t_i \leq t < t_{i+1} \\ (1 - e^{-\beta\cdot(t_i - t_{i-1})}) \cdot X_i & \end{cases}. \quad (31)$$

and its weighted number of samples is $N_j = 1$. Thus, TEMA is hardly simpler than UTEMA since it also requires the calculation of exponential functions. The equations fit the Definitions (18) – (20) for the sample-specific weight functions

$$g_i(t) = \begin{cases} e^{\beta\cdot t} & i = 0 \\ (1 - e^{-\beta\cdot(t_i - t_{i-1})}) \cdot e^{\beta\cdot t} & i > 0 \end{cases}. \quad (32)$$

While EMA has only two different sample-specific weight functions, those of TEMA may all be different. The contributions are also sample-specific:

$$C_i = \begin{cases} \frac{1}{\beta} & i = 0 \\ \frac{1-e^{-\beta\cdot(t_i - t_{i-1})}}{\beta} & i > 0 \end{cases}. \quad (33)$$

Nevertheless, the weight functions of all samples reveal the same memory $M = \frac{1}{\beta}$ which equals the one of UTEMA. TEMA's bias towards some samples is more severe than the one of EMA because it does not vanish over time. Its impact is illustrated in Figure 2(c). Sample sizes are 0 after a short inter-sample time of $0.1 \cdot \Delta t$ and 1 after a long inter-sample time of $1.9 \cdot \Delta t$. While UTEMA yields average values converge to 0.5 most of the time, TEMA's average values continuously increase as large samples have a larger contribution than small samples in this process. We also simulated a Poisson arrival process over $10^6 \cdot \Delta t$ time with rate $\lambda = \frac{1}{\Delta t}$ and set the sample size X_i to the value of the preceding inter-arrival time divided by Δt. We averaged the obtained time-dependent average values over time. UTEMA yields 1.11, 1.05, and 1.02 for $M \in \{\Delta t, 4 \cdot \Delta t, 10 \cdot \Delta t, 25 \cdot \Delta t\}$ while corresponding values for TEMA are 1.80, 1.91, and 1.96. Thus, the bias is significant and even increases with larger memory.

3.3 Summary

Table 1 summarizes properties of considered MAs.

4. ANALYSIS OF UEMA

We illustrate the impact of UEMA's memory on the accuracy and timeliness of obtained averages. Accuracy addresses the deviation of UEMA's computed average from the true mean μ of an observed stationary sample process $(X_i)_{0 \leq i < \infty}$. Timeliness addresses UEMA's ability to early reflect changes regarding μ in the observed process.

Table 2: Averaged squared deviation of computed averages A_i from the known sample mean μ.

	σ^2					
$M(\Delta t)$	1	3	10	30	100	300
3	0.1999	0.5997	1.9990	5.9969	19.9896	59.9687
10	0.0527	0.1580	0.5266	1.5798	5.2655	15.7966
30	0.0170	0.0511	0.1734	0.5112	1.7039	5.1116
100	0.0051	0.0152	0.0508	0.1525	0.5082	1.5246
300	0.0017	0.0051	0.0171	0.0514	0.1712	0.5136
1000	0.0005	0.0016	0.0053	0.0158	0.0527	0.1580
CumMean	0.0000	0.0000	0.0001	0.0004	0.0012	0.0036

4.1 Impact of Memory on Accuracy

We generate $n = 10^6$ normally distributed random variables X_i with mean $\mu = 0$ and variance $\sigma^2 \in \{1, 3, 10, 30, 100, 300\}$ and track them by UEMA with a memory of $M \in \{3, 10, 30, 100, 300, 1000\} \cdot \Delta t$. Table 2 shows the arithmetic mean of the squared deviations of the averages from the observed random variable's mean μ: $dev^2(n) = \frac{1}{n} \cdot \sum_{0 \leq i < n}(A_i - \mu)^2$. The accuracy of the estimate depends on the variance of the observed process and UEMA's memory. It can be explained as follows. According to Equations (2) – (4) and (13), UEMA's average value A_j is the sum S_j of weighted, independent random variables multiplied by the scalar value $\frac{1}{N_j}$. Therefore, the variance of A_j can be calculated as

$$VAR[A] = \frac{\sum_{0 \le i < \infty} a^{2 \cdot i} \cdot \sigma^2}{(\sum_{0 \le i < \infty} a^i)^2} = \frac{(1-a)^2}{1-a^2} \cdot \sigma^2 = \frac{\sigma^2}{2 \cdot \frac{M}{\Delta t} - 1} \quad (34)$$

which explains the observation in our experiment.

Equation (34) helps to find a minimum smoothing factor a if the variance σ^2 of the samples is roughly known. We assume that the averaged values A_j are distributed according to a normal distribution. With $E[A] = \mu$ and $VAR[A] = \frac{1-a}{1+a} \cdot \sigma^2$ we define $Y = \frac{A-\mu}{\sqrt{\frac{1-a}{1+a} \cdot \sigma^2}}$ which is distributed according to a standard normal distribution[2]. Therefore, we get

$$P(-z_{1-\frac{\alpha}{2}} \quad \le Y \le \quad z_{1-\frac{\alpha}{2}}) = 1 - \alpha \quad (35)$$

$$P(\mu - \delta(a,\alpha) \quad \le A \le \quad \mu + \delta(a,\alpha)) = 1 - \alpha \quad (36)$$

with $z_{1-\frac{\alpha}{2}}$ being the $(1-\frac{\alpha}{2})$-quantile of the standard normal distribution and $\delta(a,\alpha) = z_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{1-a}{1+a} \cdot \sigma^2}$. We conclude that A deviates at most δ from the true mean μ with probability $1-\alpha$. This derivation helps to understand the accuracy of UEMA depending on a, but cannot calculate the accuracy when random variables with unknown variance σ^2 are observed. A solution is to assume orders of magnitude for σ^2.

We use this derivation to choose a lower bound $a_{min}(\alpha, \varepsilon)$ for a such that a maximum error ε can be achieved with probability $1-\alpha$:

$$a_{min}(\alpha, \varepsilon) \ge \frac{1 - \left(\frac{\varepsilon}{\sigma \cdot z_{1-\frac{\alpha}{2}}}\right)^2}{1 + \left(\frac{\varepsilon}{\sigma \cdot z_{1-\frac{\alpha}{2}}}\right)^2}. \quad (37)$$

Table 3 illustrates the accuracy of that approach for 10^7 normally distributed random variables and compares it with a standard smoothing factor of $a = 0.9$.

Table 3: Percentage ρ of UEMA averages deviating at most $\varepsilon = 1$ from the actual mean $\mu = 0$ depending on smoothing factor a. Smoothing factor a_{min} is chosen for $\alpha = 0.1$ ($z_{1-\frac{\alpha}{2}} = 1.645$) according to Equation (37).

σ^2	1	3	10	30	100	300
$a_{min}(\alpha, \varepsilon)$	0.4579	0.7795	0.9283	0.9755	0.9926	0.9975
$\rho(a_{min}(\alpha, \varepsilon))$	0.8990	0.8989	0.8990	0.8902	0.8991	0.9001
$\rho(a = 0.9)$	1.0000	0.9881	0.8319	0.5740	0.3371	0.1987

Equation (37) helps to estimate probabilities with a certain accuracy. Probabilities p can be determined by counting $X = 1$ if a sample fulfills a certain condition, and $X = 0$ otherwise. This yields a Bernoulli process with variance $\sigma^2 = (1-p) \cdot p \le 0.25$ which may be used to determine an appropriate smoothing factor. Thus, $a_{min} = 0.99971$ may be used to estimate probabilities with accuracy of $\varepsilon = 0.01$ and error probability of $\alpha = 0.1$.

4.2 Impact of Memory on Timeliness

To illustrate the ability of UEMA to reveal changed process behavior, we simulate $n+1 = 10001$ samples of normally distributed random variables X_i. The mean of X_i increases over time and is set to $E[X_i] = \frac{i}{n}$, $0 \le i \le n$ while the variance $\sigma^2 = 1$ remains stable. We performed this experiment $n_{runs} = 10000$ times to calculate average $\overline{A_i}$ and the empirical variance $S^2(A_i)$ of the average values A_i. Table 4 shows these values estimated after $i \in \{3333, 6666, 10000\}$ steps by UEMA for different memory and for CumMean.

Table 4: Samples with increasing expectations $E[X_i]$. Average ($\overline{A_i}$ and empirical variance $S^2(A_i)$ of average values A_i for UEMA with different memory and for CumMean (CM).

i	3333		6666		10000	
$M(\Delta t)$	$\overline{A_i}$	$S^2(A_i)$	$\overline{A_i}$	$S^2(A_i)$	$\overline{A_i}$	$S^2(A_i)$
1	0.3355	0.9987	0.6773	1.0051	0.9886	1.0128
3	0.3297	0.1983	0.6680	0.2021	0.9949	0.2022
10	0.3320	0.0533	0.6648	0.0530	0.9975	0.0521
30	0.3305	0.0174	0.6623	0.0169	0.9964	0.0166
100	0.3233	0.0052	0.6557	0.0051	0.9899	0.0049
300	0.3032	0.0017	0.6361	0.0017	0.9702	0.0016
1000	0.2458	0.0005	0.5672	0.0005	0.9901	0.0005
3000	0.1972	0.0003	0.4475	0.0002	0.7370	0.0002
10000	0.1762	0.0003	0.3701	0.0002	0.5820	0.0001
CM	0.1670	0.0003	0.3334	0.0002	0.5001	0.0001
$E[X_i]$	0.3333	-	0.6666	-	1.0000	-

The table shows that the averaged averages $\overline{A_i}$ approximate the configured expectations $E[X_i]$ well for small memory and clearly underestimate them for larger memory. Best values are obtained in this specific experiment for a memory of at most $M = 100 \cdot \Delta t$. We also observe that the sample variance $S^2(A_i)$ depends on the memory M but not on the index i. The latter is due to the fact that all X_i have the same variance σ^2 so that the expectation of the sample variance can be approximated by $VAR[A_i] = \frac{1}{2 \cdot \frac{M}{\delta t} - 1} \cdot \sigma^2$ which is well approximated by the values in the table. As A_i computed with small memory, e.g., $M = 10 \cdot \Delta t$ or smaller, reveal a large variance, they often deviate from their average $\overline{A_i}$ and are only little reliable. Thus, there is a tradeoff between accuracy and timeliness that can be controlled by UEMA's memory.

To guarantee timeliness of computed averages, the smoothing factor a must be low enough. The last m samples contribute an overall weight of $\sum_{0 \le i < m} a^i = \frac{1-a^m}{1-a}$ to the average while the average contains an overall weight of at most $\sum_{0 \le i < \infty} a^i = \frac{1}{1-a}$. To limit the influence of samples older than m time steps to a fraction γ, $\frac{1-a^m}{1-a} \ge \frac{1-\gamma}{1-a}$ must hold, i.e., $a \le \sqrt[m]{\gamma}$ must be met. Conversely, for a given a, the impact of samples older than $\lceil \frac{\ln(\gamma)}{\ln(a)} \rceil$ time steps is limited to γ.

5. MOVING HISTOGRAMS (MH)

A histogram partitions the sample range into k intervals and associates with each of them a counter $bin(i)$, $0 \le i < k$. We denote their lower and upper bound by $l(i)$ and $u(i)$. We define that the lower bound is part of the preceding interval and that the upper bound is part of the considered interval. Left- and rightmost intervals are extended towards $\pm\infty$. All bins are initialized with zero. At sample arrival, the corresponding bin of a cumulative histogram (CumHist) is incremented by 1. The relative frequency for samples in a certain interval i can be calculated by

$$h(i) = \frac{bin(i)}{\sum_{0 \le j < k} bin(j)}. \quad (38)$$

While other relative frequencies such as $h(X \le x)$ can also be determined by MAs, histograms allow the approximation of quantiles. The p-quantile Q_p is the infimum of values x for which $P(X \le x) \le p$ holds. Histograms can approximate it by

$$\widehat{Q_p} = u(i) : \sum_{0 \le j < i} h(j) < p \le \sum_{0 \le j \le i} h(j). \quad (39)$$

[2] The approach taken is similar but not equal to the estimation of confidence intervals for the mean of independent samples. In our case, consecutive average values A_j are highly correlated so that the variance cannot be derived from samples. We solve that problem by assuming an appropriate value for σ^2.

We adapt the concept of MAs to histograms. Moving histograms (MHs) can provide time-dependent quantiles. We presented an application in [12]. In the following, we discuss MHs on the base of UEMA and UTEMA.

5.1 MHs for Evenly Spaced Time Series

We present the unbiased exponential moving histogram (UEMH) which extends UEMA. When a new sample arrives, all bins are devaluated by the smoothing factor a. Afterwards, the bin associated with the new sample is incremented by 1. Thereby, the contribution of older samples to the histogram decreases. Relative frequencies are determined according to Equation (38). A single bin corresponds to UEMA's S_j and the sum of all bins to UEMA's N_j. Properties like contribution C, memory M, and delay D of UEMA also apply.

This straightforward adaptation requires high multiplication effort. As an alternative, devaluation may be omitted and $\frac{1}{a^i}$ may be used as increment for sample X_i instead of 1. This causes numerical instability as $\frac{1}{a^i}$ rises exponentially. A compromise is to avoid that increments exceed a value η. To that end, we devaluate bins only after $n_{dev} = \lceil \frac{-\ln(\eta)}{\ln(a)} \rceil$ steps by a factor $\frac{1}{a^{n_{dev}}}$ and choose $a^{i \mod n_{dev}}$ as increment for X_i.

5.2 MHs for Unevenly Spaced Time Series

The unbiased time-exponential moving histogram (UTEMH) extends UTEMA. In contrast to UEMH, bins are devaluated by $e^{-\beta\cdot(t_i - t_{i-1})}$ instead of a when X_i arrives. Similarly, increments of size $e^{\beta\cdot(t_i-t_0)}$ may be used instead of $\frac{1}{a^i}$ to avoid devaluation of all bins. To avoid numerical problems, bins are devaluated after $t_{dev} = \frac{\ln(\eta)}{\beta}$ time by $e^{-\beta\cdot t_{dev}}$ and increments $e^{\beta\cdot((t_i-t_0) \mod t_{dev})}$ are chosen.

Table 5: 10%-quantiles of UEMH with different memory and cumulative histogram (CH) for samples with increasing expectations $E[X_i]$.

i	3333		6666		10000	
$M(\Delta t)$	$\overline{Q_{10\%,i}}$	$\widehat{Q_{10\%,i}}$	$\overline{Q_{10\%,i}}$	$\widehat{Q_{10\%,i}}$	$\overline{Q_{10\%,i}}$	$\widehat{Q_{10\%,i}}$
1	-0.27	-0.89	-0.06	-0.58	+0.18	-0.36
3	-0.75	-0.92	-0.43	-0.55	-0.09	-0.29
10	-0.89	-0.92	-0.53	-0.55	-0.25	-0.26
30	-0.90	-0.92	-0.57	-0.57	-0.29	-0.28
100	-0.93	-0.94	-0.60	-0.60	-0.29	-0.29
300	-0.97	-0.97	-0.63	-0.63	-0.31	-0.31
1000	-1.03	-1.03	-0.71	-0.71	-0.38	-0.39
3000	-1.08	-1.08	-0.85	-0.85	-0.57	-0.58
10000	-1.11	-1.11	-0.93	-0.39	-0.75	-0.75
CH	-1.12	-1.11	-0.97	-0.97	-0.83	-0.83
$Q_{10\%,i}$	-0.949		-0.615		-0.282	

5.3 Timeliness of UEMH and CumHist

We illustrate the timeliness of UEMH with different memory M and cumulative histograms. We perform the same experiment as in Section 4.2 and set up histograms with equal-size ranges between -3 and 1 of width 0.1. Table 5 shows estimates of 10%-quantiles after i samples. They are computed as average values $\overline{Q_{10\%,i}}$ of 10%-quantiles gained from $n_{runs} = 100$ runs or as 10%-quantiles $\widehat{Q_{10\%,i}}$ based on the aggregated histogram information after sample i from all the n_{runs} different runs which is closer to the true value of the 10% quantile due to reduced variance. However, we see that both methods yield very similar results for $M = 30 \cdot \Delta t$ or larger. The analytical values are given on the bottom of the table. They are well approximated by both methods with memories between 30 and $300 \cdot \Delta t$. The 10% quantiles derived from a single run significantly fluctuate, but the values for $\overline{Q_{10\%,i}}$ show that they yield the right values on average. The quantiles estimated with cumulative histograms increase only slowly over time and clearly underestimate the analytical values. Smaller memory for UEMH cannot hold enough data for sufficiently accurate calculation of quantiles. MHs with larger memory are too much influenced by older samples which were generated with lower mean and cause lower estimates. Thus, there is also a tradeoff between timeliness and accuracy for MHs.

6. TIME-DEPENDENT RATE MEASUREMENT (TDRM)

A rate denotes an average number of samples per time, possibly weighted by their size. A time-dependent rate reflects mainly the recent past of the observed process. We present various techniques for TDRM and provide a comparison. Among the considered methods, TDRM-UTEMA is new and excels through timeliness, ease of configuration, and the fact that its measured rates are continuous with regard to configured memory.

6.1 TDRM Methods

We suggest a framework for the definition of TDRM methods and present five different instantiations.

6.1.1 A Framework for TDRM Methods

A time-dependent rate R_t at time t may be determined by

$$R_t = \begin{cases} \frac{S_t}{T_t} & T_t > 0 \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

$$T_t = \int_0^t g(\tau - t)d\tau \tag{41}$$

where S_t is the weighted sample sum, taken from some MA method, and T_t is the weighted measurement interval which is computed analogously to S_t. Measured rates depend on a time scale which is the duration over which samples are considered for rate computation. In the presented definition the time scale is inherited from the memory M of the applied MA method. Also the concepts contribution C and delay D are inherited.

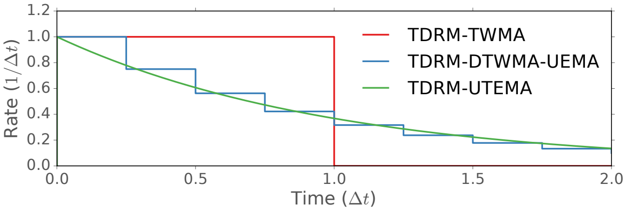


Figure 4: Rate impulses for TDRM-{TWMA, DTWMA-UEMA, UTEMA} with $t \to \infty$ used for calculation of T_t.

6.1.2 TDRM with Time Window Moving Average (TDRM-TWMA)

TDRM-TWMA calculates the sample sum S_t according to Equation (24). The corresponding weighted measurement interval is $T_t = \min(t, W)$ under the assumption that the measurement process starts at $t = 0$. The memory is

$M = W$ and the delay is $D = \frac{W}{2} = \frac{M}{2}$. As S_t may be zero, measured rates may be zero. TDRM-TWMA requires temporary storage of the samples within the measurement window and a timer indicating when the next sample leaves it. Therefore, the computation memory needed for TDRM-TWMA scales with the configured memory M and the rate of the process to be measured.

Figure 4 visualizes rate impulses for different TDRM methods. TDRM-TWMA generates a rate $\frac{1}{M}$ over a duration M for every observed sample of unit size. This rate can be considered as an impulse and the superposition of impulses from all measured samples (scaled by their size) yields the measured rate. The figure depicts rate impulses for various TDRM methods. They have the same memory M but contribute with a different time-dependent rate impulse to an overall measured rate. The impulses have an integral of 1 as $t \to \infty$ is assumed for computation of T_t. Superposition of such impulses ensures that the average of the overall measured rate approximates the sum of the observed sample sizes divided by the duration of the observation interval.

6.1.3 TDRM with Disjoint Time Windows Moving Average (TDRM-DTWMA)

TDRM-DTWMA computes rates for disjoint time windows of duration W. It uses the weighted sum S_t for DTWMA in Equation (26) and $T_t = W$. Memory is $M = W$ and average delay is $\overline{D} = M$. As S_t may be zero, measured rates may be zero. In contrast to TDRM-TWMA, TDRM-DTWMA does not necessarily require storage of samples. It yields the same rate impulse like TDRM-TWMA, but the impulse for a sample may become visible only in the next measurement window.

6.1.4 TDRM with DTWMA and (U)EMA (TDRM-DTWMA-(U)EMA)

TDRM-DTWMA-(U)EMA calculates rates according to TDRM-DTWMA and smoothes them with (U)EMA. It is used in [9] for dequeue rate estimation and requires two parameters: the window size W and the smoothing parameter a. The resulting memory is $M = \frac{W}{1-a}$ and the average delay is $\overline{D} = \frac{W}{2} + \frac{a \cdot W}{1-a} + \frac{W}{2} = M$. An advantage of this method over TDRM-DTWMA is that the effect of measured samples becomes visible after at most W instead of M time. Furthermore, samples contribute with vanishing degree to all future measured rates instead of only to the measured rate of the following measurement window.

The rate impulse starts with rate $\frac{1-a}{W} = \frac{1}{M}$ and is reduced by a factor of a for consecutive measurement intervals of duration W. Again, the rate impulse of a sample may become visible only at the beginning of the next measurement interval. This takes at most $W = M \cdot (1-a)$ time and makes TDRM-DTWMA-(U)EMA react faster than TDRM-DTWMA although they both exhibit the same average delay. Unlike TDRM-DTWMA-EMA, TDRM-DTWMA-UEMA does not suffer from a bias towards the measured rate of the first measurement window.

6.1.5 TDRM with UTEMA (TDRM-UTEMA)

TDRM-UTEMA leverages the weighted sum S_t of UTEMA according to Equation (29) and uses the weighted time

$$T_t = \int_0^t e^{-\beta \cdot (t-\tau)} d\tau = \frac{1}{\beta} \cdot (1 - e^{-\beta \cdot t}). \tag{42}$$

The memory and delay of TDRM-UTEMA are $M = D = \frac{1}{\beta}$. The rate impulse of TDRM-UTEMA is an exponentially decreasing function. In Figure 4 it is visualized for $T_t = \frac{1}{\beta}$, i.e., idealized for $t \to \infty$, and resembles the rate impulse of TDRM-DTWMA-UEMA. In fact, TDRM-UTEMA can be viewed as the limit of TDRM-DTWMA-UEMA for decreasing window sizes W.

The advantage of TDRM-UTEMA over TDRM-DTWMA-UEMA is its immediate reaction and the need for only a single parameter β. The advantage of TDRM-DTWMA-UEMA is its computational efficiency as it does not require the computation of exponential functions.

6.1.6 TDRM with UTEMA and Continuous Packet Arrivals (TDRM-UTEMA-CPA)

In [8], the following recursion formula has been applied for online rate computation:

$$R_{t_i} = \begin{cases} \frac{X_0}{t_0} & i = 0 \\ e^{-\beta \cdot (t_i - t_{i-1})} \cdot R_{t_{i-1}} + & \text{otherwise} \\ (1 - e^{-\beta \cdot (t_i - t_{i-1})}) \cdot \frac{X_i}{t_i - t_{i-1}} & \end{cases} . \tag{43}$$

The start of the measurement period is at $t = 0$. Later in this section we show that this recursive formula essentially implements TDRM-UTEMA with the assumption that packets continuously arrive during their preceding inter-arrival time instead of arriving instantly at their actual arrival time.

6.2 Comparison of TDRM Methods

We first visualize the results of TDRM-{TWMA, DTWMA, DTWMA-UEMA, UTEMA} for burst arrivals with equal inter-arrival times. Then we measure a Poisson arrival process using different memory. Finally, we show that TDRM-UTEMA-CPA can be derived from TDRM-UTEMA and point out its shortcomings.

6.2.1 Measuring a Burst

The bottom line of Figure 5 shows a burst of equal-size packets arriving with equal inter-arrival times at a rate of $\lambda = \frac{1}{\Delta t}$. Before and after the burst the arrival rate is zero. The figure illustrates the measured rates for the above mentioned TDRM methods, each configured with a memory of $M = 10 \cdot \Delta t$.

TDRM-TWMA produces a step function that first linearly increases, reaches a measured rate of $\frac{1}{\Delta t}$ after $M = 10 \cdot \Delta t$ time, stays constant for a while, then linearly decreases, and reaches zero again after M time. The method exhibits a high timeliness as the rate increases shortly after the arrival of the first packet and returns to zero shortly after the arrival of the last packet.

TDRM-DTWMA reveals a positive rate only with the start of the next measurement window after the arrival of the first packet and captures the full rate only one measurement window later. The end of the burst is reflected late by TDRM-DTWMA's measured rate.

TDRM-DTWMA-UEMA also yields a step function. Its measured rate reflects the beginning of the burst earlier than TDRM-DTWMA since it uses a shorter measurement window. However, it takes some time to approach the full observed rate of $\frac{1}{\Delta t}$ because it does not completely forget about the past when no packets arrived. In a similar way, the measured rate geometrically decreases after the arrival of the last

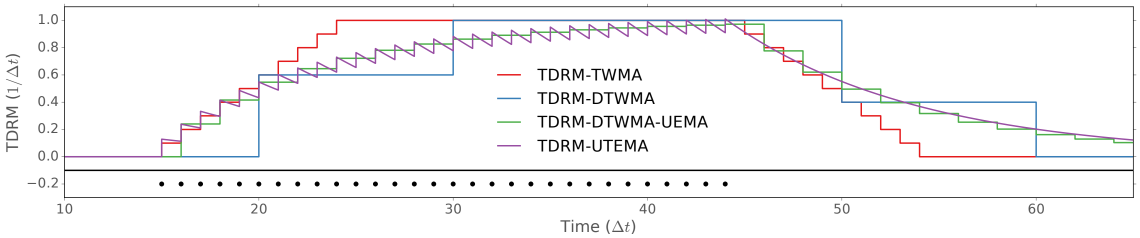


Figure 5: Rate measurement of burst arrivals with constant inter-arrival times Δt. Packet arrivals are shown on the bottom of the figure. The TDRM methods are configured with a memory of $M = 10 \cdot \Delta t$. TDRM-DTWMA-UEMA uses a window of $W = 2 \cdot \Delta t$ and $a = \frac{4}{5}$.

packet. Therefore, it takes long to approach zero. We used a measurement window of $W = 2 \cdot \Delta t$ for the rate curve in the figure. A measurement window of $W = 2.5 \cdot \Delta t$ alternately covers 2 or 3 arrivals which imposes an oscillating behavior on the obtained rates although the observed process has constant rate. This is a general, undesired artifact of window-based TDRM methods.

The rate measured by TDRM-UTEMA jumps with every packet arrival and exponentially decreases in the absence of new samples. Therefore, its shape at large resembles the one measured by TDRM-DTWMA-UEMA, but TDRM-UTEMA does not exibit the above mentioned artefact.

6.2.2 Measuring a Poisson Process

We consider a Poisson arrival process with an arrival rate of $\lambda = \frac{1}{\Delta t}$ and equal-size samples over a duration of $10^6 \cdot \Delta t$. A cutout between $40 \cdot \Delta t$ and $240 \cdot \Delta t$ is illustrated in Figure 6(a). Figures 6(b)–6(e) illustrate time-dependent rates of this process measured by TDRM-{TWMA, DTWMA, DTWMA-UEMA, UTEMA} with a memory of $M = 20 \cdot \Delta t$ and $M = 40 \cdot \Delta t$ as well as the rate difference between these curves.

TDRM-TWMA's measured rate in Figure 6(b) is a step function and changes whenever a new sample arrives or an old sample leaves the measurement window. Frequently arriving samples cause rising or high rates while frequently leaving samples cause falling or low rates. The curves measured with the longer memory of $M = 40 \cdot \Delta t$ are influenced by the same arriving samples but different leaving samples compared to the curves measured with shorter memory. Therefore, the memory has a clear and non-continuous impact on measured TDRM-TWMA's rates. The resulting rate difference is also shown in the figure. We calculate the average absolute rate difference and denote it by R^{abs}_{diff}. It is given in the captions of the figures and it is relatively high for TDRM-TWMA. The variance of TDRM-TWMA's rate curves is rather high because it computes the rate only from the small number of samples within its measurement window. To quantify this observation, we compute the coefficients of variation $c_{var}(M)$ of the rate curves and also report them in the captions of the figures.

The rate curves for TDRM-DTWMA in Figure 6(c) suffer from the same problems as TDRM-TWMA which is quantified by R^{abs}_{diff} and $c_{var}(M)$. They behave similarly as those for TDRM-TWMA but are clearly delayed. Some low (high) values of TDRM-TWMA are suppressed, e.g., between $t = 160 \cdot \Delta t$ and $t = 180 \cdot \Delta t$ because TDRM-DTWMA's few discrete measurement windows comprise both low- and high-frequent arrivals.

We consider TDRM-DTWMA-UEMA configured with a window of size $5 \cdot \Delta t$. Its rates are reported in Figure 6(d) and are represented by step functions. TDRM-DTWMA-UEMA's rates measured with $M = 20 \cdot \Delta t$ and $M = 40 \cdot \Delta t$ exhibit significantly less difference compared to TDRM-TWMA and TDRM-DTWMA because TDRM-DTWMA-UEMA is continuous with regard to the configured memory. The measured rates reveal a lower coefficient of variation because their calculation takes all previous samples into account.

Rates measured by TDRM-UTEMA are illustrated in Figure 6(e). They jump at each packet arrival and exponentially decrease in between. They are very similar to those measured by TDRM-DTWMA-UEMA which exhibit geometric decay in the absence of samples. TDRM-UTEMA's rates are also continuous with regard to memory. TDRM-UTEMA exhibits the least average difference R^{abs}_{diff} between rates measured with different memory. Its rate computation also respects all past samples and its rates reveal the least variance among all TDRM methods.

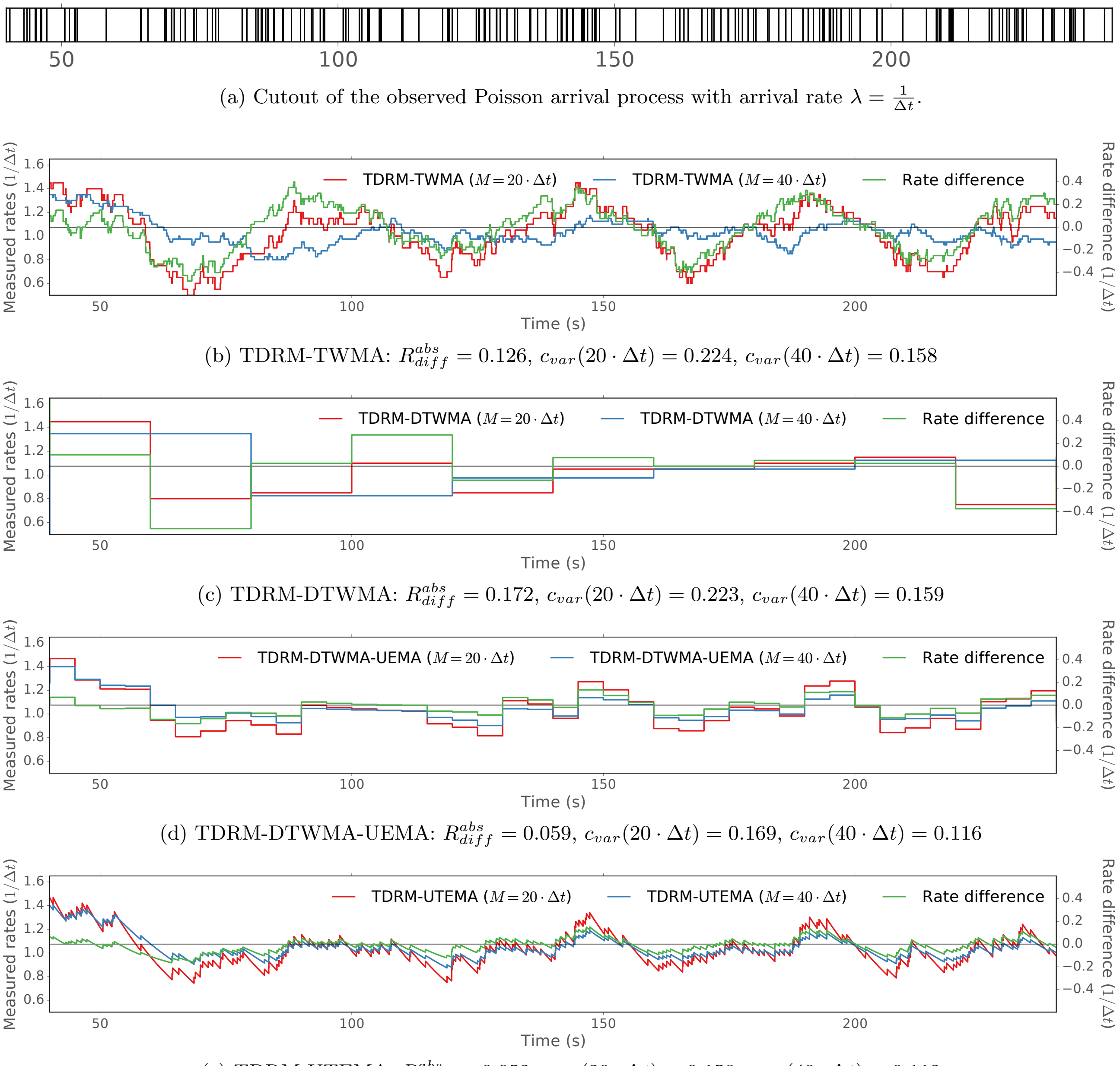
The variance of measured rates decreases for all TDRM methods with increasing memory. However, the variance for TDRM-TWMA and TDRM with $M = 40 \cdot \Delta t$ is about as large as the variance for TDRM-DTWMA-UEMA and TDRM-UTEMA with $M = 20 \cdot \Delta t$.

6.2.3 Comparison of TDRM-UTEMA-CPA and TDRM-UTEMA

We derive the recursion formula in Equation (43) to point out its connection with TDRM-UTEMA. We assume that packets X_i continuously arrive with rate $\frac{X_i}{t_i - t_{i-1}}$ during their preceding inter-arrival time and partial packets are already devaluated with passing time τ by $e^{-\beta \cdot (t_i - \tau)}$, similar to Equation (29). As a result, the remaining packet size at t_i is the modified sample size

$$\begin{aligned} X_i^* &= \int_{t_{i-1}}^{t_i} \frac{X_i}{t_i - t_{i-1}} \cdot e^{-\beta \cdot (t_i - \tau)} d\tau \\ &= \frac{X_i}{\beta \cdot (t_i - t_{i-1})} \cdot (1 - e^{-\beta \cdot (t_i - t_{i-1})}). \qquad (44) \end{aligned}$$

An exception is the first modified sample whose size is computed as $X_0^* = \frac{X_0}{\beta \cdot t_0}$. Application of TDRM-UTEMA to these modified samples yields TDRM-UTEMA-CPA's recursion formula in Equation (43). More specifically, the sample sum in Equation (29) is computed based on X_i^* instead of X_i and $T_t = \frac{1}{\beta}$ is taken as weighted time which is exact for $t \to \infty$. Therefore, TDRM-UTEMA-CPA has only an initial bias due to the different computation of X_0^*, but does not exhibit a persistent bias due to its consistency with TDRM-UTEMA. This is not obvious because Equation (43) looks at first sight like an application of TEMA to short-term rates



(a) Cutout of the observed Poisson arrival process with arrival rate $\lambda = \frac{1}{\Delta t}$.

(b) TDRM-TWMA: $R^{abs}_{diff} = 0.126$, $c_{var}(20 \cdot \Delta t) = 0.224$, $c_{var}(40 \cdot \Delta t) = 0.158$

(c) TDRM-DTWMA: $R^{abs}_{diff} = 0.172$, $c_{var}(20 \cdot \Delta t) = 0.223$, $c_{var}(40 \cdot \Delta t) = 0.159$

(d) TDRM-DTWMA-UEMA: $R^{abs}_{diff} = 0.059$, $c_{var}(20 \cdot \Delta t) = 0.169$, $c_{var}(40 \cdot \Delta t) = 0.116$

(e) TDRM-UTEMA: $R^{abs}_{diff} = 0.052$, $c_{var}(20 \cdot \Delta t) = 0.158$, $c_{var}(40 \cdot \Delta t) = 0.112$

Figure 6: Poisson arrival process and time-dependent rates measured by various TDRM methods with $M = 20 \cdot \Delta t$ and $M = 40 \cdot \Delta t$; coefficient of variations (c_{var}) of and average deviations R^{abs}_{diff} between the two curves are given in the captions.

$R^*_i = \frac{X_i}{t_i - t_{i-1}}$ computed at each packet arrival, and TEMA exhibits both an initial and a persistent bias.

After the initial bias towards X^*_0 has vanished, TDRM-UTEMA-CPA yields rates for arrival instants that are slightly lower than those of TDRM-UTEMA. While TDRM-UTEMA yields a piecewise exponentially decaying function, TDRM-UTEMA-CPA updates rates only at arrival instants and leads to a step function. The missing decay during inter-arrival times can overestimate rates over time. To quantify this effect, we measured the rates of a Poisson process ($c_{var} = 1.0$) and an arrival process whose inter-arrival times have a coefficient of variation of $c_{var} = 2.0$. The processes have an arrival rate of $\lambda = \frac{1}{\Delta t}$ and take $10^6 \cdot \Delta t$ time. Table 6 shows average rates measured by TDRM-UTEMA and TDRM-UTEMA-CPA with a memory of $M \in \{10, 100\} \cdot \Delta t$. TDRM-UTEMA does not overestimate rates while the overestimation through TDRM-UTEMA-CPA is significant for short memory M and the arrival process with highly varying inter-arrival times.

Table 6: Average rates measured by TDRM-{UTEMA,UTEMA-CPA}.

	$c_{var}(A) = 1.0$		$c_{var}(A) = 2.0$	
Memory M	$10 \cdot \Delta t$	$100 \cdot \Delta t$	$10 \cdot \Delta t$	$100 \cdot \Delta t$
TDRM-UTEMA	1.000	1.000	1.000	1.000
TDRM-UTEMA-CPA	1.048	1.005	1.186	1.020

6.3 Summary

We have proposed a framework for TDRM. We considered four methods from literature and a novel one: TDRM-UTEMA. Only TDRM-TWMA and TDRM-UTEMA immediately reflect rate changes in measured time-dependent rates. In contrast, TDRM-DTWMA and TDRM-DTWMA-UEMA take some time until rate changes become visible. Thereby, TDRM-DTWMA-UEMA uses shorter measurement windows than TDRM-DTWMA and exponential smoothing so that it can react faster to rate increases than TDRM-DTWMA. For small measurement windows, rates measured by TDRM-DTWMA-UEMA converge to those measured by TDRM-UTEMA. While TDRM-TWMA and TDRM-DTWMA yield measurement curves with high variance, TDRM-UTEMA and TDRM-DTWMA-UEMA exhibit clearly lower variance when being configured with the same memory because they average over all past samples. While time-dependent rates measured by TDRM-DTWMA-UEMA and TDRM-UTEMA are continuous with regard to configured memory, rates measured by TDRM-TWMA and TDRM-DTWMA significantly depend on the configured memory. This makes TDRM-TWMA and TDRM-DTWMA more difficult to interpret and use. The novel TDRM-UTEMA is the only of these studied methods (1) whose measured rates are continuous with regard to memory and (2) immediately react to rate changes, (3) which can be configured by a single parameter, and (4) which cannot not produce window-based artifacts. Therefore, its measured rates depend least on the chosen memory. Nevertheless, the two-parametric TDRM-DTWMA-UEMA can possibly serve as a less computation-intensive approximation of TDRM-UTEMA. We showed that TDRM-UTEMA-CPA is a variant of TDRM-UTEMA but may overestimate rates.

7. CONCLUSION

We have presented a framework for the definition of moving averages (MAs) including performance metrics like memory and delay. We presented several MA variants for evenly and unevenly spaced time series, showed that they fit well into that framework, and demonstrated that some of them have a bias. We proposed the unbiased exponential MA (UEMA) and the unbiased time-exponential MA (UTEMA) as novel MA methods that avoid such a bias. We configured MA methods such that they revealed the same memory and produced comparable results. Our analysis of UEMA showed that the memory allows for a tradeoff between accuracy and timeliness when mean values are determined. We also suggested some equations that help to choose appropriate smoothing parameters when UEMA should provide average values with a certain accuracy. We discussed moving histograms (MHs) as an extension of MAs and showed how they may be used to determine quantiles. Finally, we extended the framework to time-dependent rate measurement (TDRM). We embedded four existing TDRM methods in that framework and suggested TDRM-UTEMA as a novel method that excels by its timeliness, ease of configuration, and by the fact that its measured rates continuously depend on the configured memory. We performed experiments to illustrate and compare these TDRM methods and pointed out their pros and cons.

We presented a demo of all methods discussed in this work at [18] and made the source code available at [19].

This work focused on online measurement, i.e., on the assumption that future samples of the measured process are unknown. It has applicability in self-adaptive systems. The work may be extended to offline measurement so that time series can be smoothed while taking both past and future samples into account when calculating averages, histograms, or time-dependent rates.

8. REFERENCES

[1] Information Sciences Institute, “RFC793: Transmission Control Protocol,” September 1981.

[2] G. Jenkins and D. Watts, *Spectral analysis and its applications.* Holden-Day, 1968.

[3] R. M. Chiulli, *Quantitative Analysis: An Introduction.* CRC Press, 1999.

[4] A. Eckner, “Algorithms for Unevenly Spaced Time Series: Moving Averages and Other Rolling Operators,” Apr. 2012.

[5] E. Zivot and J. Wang, *Modelling Financial Time Series with S-Plus.* Springer, 2006.

[6] L. Burgstahler et al., “New Modifications of the Exponential Moving Average Algorithm for Bandwidth Estimation,” in *ITC Specialist Sem.*, 2002.

[7] M. Alizadeh *et al.*, “CONGA: Distributed Congestion-Aware Load Balancing for Datacenters,” in *ACM SIGCOMM*, Chicago, IL, USA, Aug. 2014.

[8] I. Stoica et al., “Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High-Speed Networks,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, Feb. 2003.

[9] R. Pan *et al.*, “RFC8033: Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem,” https://tools.ietf.org/html/rfc8033, Feb. 2017.

[10] R. Martin and M. Menth, “Improving the Timeliness of Rate Measurements,” in *GI/ITG MMB*, 2004.

[11] G. Bianchi et al., “On-Demand Time-Decaying Bloom Filters for Telemarketer Detection,” *ACM CCR*, vol. 41, no. 5, Oct. 2011.

[12] M. Menth et al., “Time-Exponentially Weighted Moving Histograms (TEWMH) for Application in Adaptive Systems,” in *IEEE Globecom*, 2006.

[13] W. D. Kelton and A. M. Law, *Simulation Modeling and Analysis.* McGraw Hill Boston, 2000.

[14] R. G. Brown and R. F. Meyer, “The Fundamental Theorem of Exponential Smoothing,” *Operations Research*, vol. 9, no. 5, pp. 673–685, 1961.

[15] C. Chatfield *et al.*, “The Holt-Winters Forecasting Procedure,” *Appl. Statistics*, vol. 27, no. 3, 1978.

[16] S. J. Hunter, “The Exponentially Weighted Moving Average,” *Jrnl. of Quality Techn.*, vol. 4, no. 18, 1986.

[17] P. E. Maravelakis *et al.*, “An EWMA Chart for Monitoring the Process Standard Deviation when Parameters are Estimated,” *Comp. Statistics & Data Analysis*, vol. 53, no. 7, 2009.

[18] M. Menth and F. Hauser, “Demo: Time Series Online Measurement for Python (TSOMpy),” in *ACM/SPEC International Conference on Performance Engineering (ICPE)*, L’Aquila, Italy, Apr. 2017.

[19] ——, “TSOMpy – Time Series Online Measurement for Python,” https://www.github.com/uni-tue-kn/TSOMpy, 2017.