CA-Smooth: Content Adaptive Smoothing of Time Series Leveraging Locally Salient Temporal Features *

Rosaria Rossini
Links Foundation
Torino country Italy
rosaria.rossini@linksfoundation.com

Silvestro Poccia
Univ. of Torino
Torino, Italy
poccia@di.unito.it

K.Selcuk Candan
Arizona State University
Tempe, Arizona, USA
candan@asu.edu

Maria Luisa Sapino
Univ. of Torino
Torino, Italy
mlsapino@di.unito.it

ABSTRACT

Imprecision and noise in the time series data may result in series with similar overall behaviors being recognized as being dissimilar because of the accumulation of many small local differences in noisy observations. While smoothing techniques can be used for eliminating such noise, the degree of smoothing that needs to be performed may vary significantly at different parts of the given time series. In this paper, we propose a content-adaptive smoothing, *CA-Smooth*, technique to reduce the impact of non-informative details and noise in time series by means of a data-driven approach to smoothing. The proposed smoothing process treats different parts of the time series according to local *information content*. We show the impact of different adaptive smoothing criteria on a number of samples from different datasets, containing series with diverse characteristics.

CCS CONCEPTS

• **Information systems** → **Temporal data**.

KEYWORDS

Time series, smoothing, salient features

*Research is supported by NSF1909555 "pCAR: Discovering and Leveraging Plausibly Causal (p-causal) Relationships to Understand Complex Dynamic Systems", NSF1827757 "Data-Driven Services for High Performance and Sustainable Buildings", NSF1610282 "DataStorm: A Data Enabled System for End-to-End Disaster Planning and Response", NSF1633381 "BIGDATA: Discovering Context-Sensitive Impact in Complex Systems", and "FourCmodeling": EU-H2020 Marie Sklodowska-Curie grant agreement No 690817.

MEDES '19, November 12–14, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-6238-2/19/11…$15.00
https://doi.org/10.1145/3297662.3365830

ACM Reference Format:
Rosaria Rossini, Silvestro Poccia, K.Selcuk Candan, and Maria Luisa Sapino. 2019. CA-Smooth: Content Adaptive Smoothing of Time Series Leveraging Locally Salient Temporal Features . In *11th International Conference on Management of Digital EcoSystems (MEDES '19), November 12–14, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3297662.3365830

1 INTRODUCTION

Filtering is a function that transforms one time series into another by altering the values of its elements in a way that takes into account their neighborhoods. The filtering operation can smooth, dampen, or accentuate fluctuations contained in the time series data [16]. Smoothing is performed through a convolution function that takes as input a time-series and a smoothing filter – the convolution operation slides the filter over the time series and for each position the overlapping values of the series and the kernel are multiplied and summed up:

Definition 1.1 (Smoothing). Given a time series $T = [t_1, t_2, ..., T_n]$ of length n, and a smoothing filter $\Phi = [\phi_1, \ldots, \phi_m]$, the convolution $T * \phi$ is defined as:

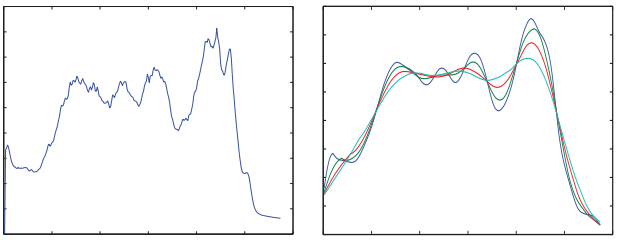
$$t_i * \Phi = \sum_k \phi_k * t_{i-k+\frac{m}{2}},$$

Where $*$ denotes the convolution operator and m is the length of the filter.

In this paper, we assume that the smoothing filter, Φ, is defined through a Gaussian filter:

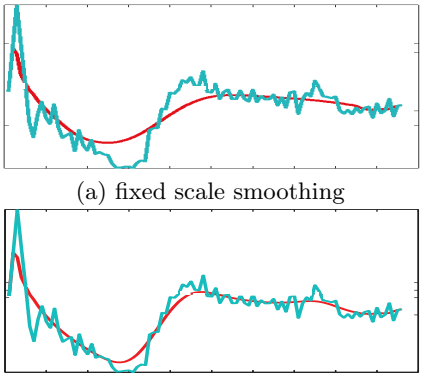
$$\phi_x = G(x - \frac{m}{2}, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x-\frac{m}{2}^2}{2\sigma^2}}.$$

The degree of the smoothing process can be controlled by varying the smoothing weights, determined by the standard deviation, σ, of the Gaussian – note that σ also defines the width of the Gaussian kernel (a kernel of length 6σ captures $\sim 99.9\%$ of the smoothing weights). In Figure 1, we see four different smoothed versions of a given series, differing in the value of σ and thus the size of the smoothing neighborhood considered: in general, the bigger the size of the smoothing filter, the more details are lost as each smoothed



(a) original time-series (b) alternative smoothings

Figure 1: Alternative smoothings of time series



(a) fixed scale smoothing

(b) content-adaptive smoothing

Figure 2: Fixed scale vs. content-adaptive smoothing

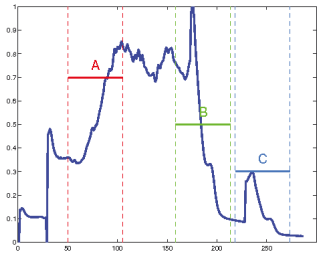


Figure 3: Three sample salient temporal features of a time series: In [4], authors recognized that salient temporal features can represent salient events on a time series; in this paper, we argue that they can also help partition a series into fragments that show different characteristics

value captures a weighted average of a larger fragment of the time series.

1.1 Proposed Approach: Content Adaptive Smoothing of Time Series, CA-Smooth

While smoothing can be used for eliminating noise, in practice, it may be difficult to determine *a priori* the amount of smoothing that needs to be applied: in fact, the amount of smoothing that needs to be performed may vary depending on what part of the time series is being considered. In particular, we argue that different parts (fragments) of the series potentially carry different amounts of redundancy and, thus, we propose to *localize* the smoothing process by (a) partitioning the series into fragments, (b) which are then independently smoothed according to their individual local information characteristics. Therefore, in this paper, we propose a content adaptive smoothing technique, *CA-Smooth*, to reduce the impact of non-informative details and noise in time series by means of a data-driven approach to local filtering. The four-step smoothing process treats the different parts of the time series differently, depending on the local *information content*:

- **Locally-salient feature detection:** In the first step, *CA-Smooth* identifies locally salient features appearing in the time series, at different scales, by means of a salient feature detection algorithm [4] (see Figure 3).
- **Fragment detection:** In the second step, the algorithm uses these locally salient features to partition the input series into *coherent* fragments.
- **Fragment characterization:** In the third step, these fragments are analyzed to identify their key properties relevant for the smoothing operation. More specifically, we consider three properties to characterize time series fragments: *entropy* and *representativeness*.
- **Content-adaptive filtering:** Finally, the identified properties of each fragment are used for associating to each time instant t a locally appropriate smoothing parameter, $\sigma(t)$, which will help preserve the critical local information, while dropping the non relevant details around time instant, t.

As visualized in Figure 2, the proposed content-adaptive *CA-smooth* algorithm treats different parts of the time series according to local the content.

1.2 Organization of the Paper

The paper is organized as follows: In the next section we discuss related works. In Section 3, we introduce the details of the proposed content adaptive smoothing strategy, *CA-Smooth*. In Section 4, we present experimental evaluations and we conclude the paper in Section 5.

2 RELATED WORK

Imprecision and noise in the time series data may result in series with similar overall behaviors being recognized as being dissimilar because of the accumulation of many small local differences in noisy observations. This potentially affects the results of (similarity based) clustering and classification algorithms. In general, the phase of noise reduction is a very important preprocessing step in time series analysis; for this reason, many techniques have been developed [5, 14, 15] to reduce and remove the noise from the signal.

Although smoothing has been recognised as a technique that can contribute to noise reduction [2], recent studies

including [8] have shown that existing unsupervised smoothing technique do not significantly contribute to clustering and classification improvement. In [8] authors show that the automated application of smoothing without domain expertise does not, on average, improve the performance of baseline classifiers. This "negative result" motivates our research: we claim that this lack of improvement is due to the fact that unsupervised smoothing does not take into account the peculiarities of the different portions of the considered time series; we therefore propose content adaptive (data driven) techniques to mediate between unsupervised and data/domain aware smoothing techniques. Time series smoothing techniques have also been proposed to prioritize end users attention: [11] proposes to smooth time series visualizations as much as possible to remove noise, while retaining large-scale structure to highlight significant deviations, and develop the ASAP analytical operator that automatically smooths streaming time series by adaptively optimizing the trade-off between noise reduction (i.e., variance) and trend retention (i.e., kurtosis).

An adaptive denoising algorithm presented in [6] consists of two steps: given a target length of $2n + 1$, the first step partitions time series into segments of $2n + 1$ points, in such a way that each segment has an overlap of $n + 1$ points with the next one. Then, the second step fits the segments with the best polynomial of order K. The two free parameters (K and n) are determined by studying the variance of the residual data. As opposed to our content adaptive method, this approach adapts the denoising parameters to the data, but it keeps them constant across the entire analysed time series, while, as it will be clearer in the next sections, we adapt the smoothing intensity to the local characteristics of fragments, whose length itself is not common to all the fragments but varies according to local characteristics.

3 CA-SMOOTH: CONTENT-ADAPTIVE TIME SERIES SMOOTHING

As described in the Introduction, in this paper, we argue that different parts of a given time series may carry different amounts of noise and redundancy and, thus, a content-adaptive smoothing process which *locally* varies the degree smoothing by (a) partitioning the series into fragments (b) which are then independently smoothed according to their individual local information characteristics, may lead to significantly better smoothing results than an inflexible strategy that applies the same smoothing filter across the entire time series. In this section, we present a content-adaptive smoothing algorithm, *CA-Smooth*. In particular, we describe the four key steps of the algorithm: (a) locally-salient feature detection, (b) fragment detection, (c) fragment characterization, and (d) content-adaptive filtering.

3.1 Locally-Salient Feature Detection

A series can be fragmented in different ways [4, 7, 9, 10]. A fixed segmentation strategy would partition the series into fragments of the same size, while an adaptive data-driven segmentation would identify fragment boundaries in a way that represents characteristics and features of the time series. Other techniques, such as CUTs [10], consider the curvature of the series and create segments in such a way that each segment has a minimal distance from the line connecting the first and the last points of the segment.

In *CA-Smooth*, we use a feature based approach: the temporal features used in the segmentation are identified using a locally-salient robust feature detection algorithm from our prior work [4]. Given a time series, T, of length n, the algorithm returns a set, S of salient features, where each salient feature, $s_i = \langle t_i, \sigma_i, descr_i \rangle \in S$, is a triple, where t_i is the center of the temporal feature, σ_i is the feature scale defining the 6σ temporal scope, $(t_i - 3\sigma_i, t_i + 3\sigma_i)$, of the feature, and $descr_i$ is a histogram of 1D gradients describing the local temporal structure. Intuitively, each of these features is significantly different from its neighborhood in the corresponding scale and therefore can be used to characterize the "key" events on the series. The algorithm uses a σ_{min} threshold to control the sizes of the smallest features identified on the series. A second parameter, τ, described in the next subsection, is used to control the lengths of the fragments. Since the number of salient features will impact the number of fragments (and thus their lengths), if the number of features is greater than $\frac{n}{\tau}$, we prune lower-intensity features among those that are temporally co-located.

3.2 Fragment Detection

As discussed above, the scope of a salient temporal feature describes a characteristic region in which an important change in the trend of the series happens. Our approach exploits this property to determine the fragments of a time series. In particular, we argue that a coherent data driven segmentation should preserve the integrity of salient features: i.e., portions of the series which can be associated to a given salient feature should ideally fall within the same fragment. This would ensure that a salient event would be represented in its entirety within the same fragment. Let T be a given time series of length n and S be the set of salient features. The algorithm considers the set S and returns a set of time series fragments, F, such that the features that cover the same region are grouped to create a cluster in a way that preserves their integrity.

It is easy to see that, one can order the interval boundaries of the salient features in S to obtain an initial fragmentation of the time series, where each fragment would be represented by the set of overlapping salient features at that region of the time series. This, however, may have two disadvantages: firstly, (a) we may end up with a lot of small fragments and (b) this process may partition the individual salient features, negatively impacting their integrity. To avoid these, given two consecutive interval boundaries, the algorithm checks if they represent a fragment with length less than τ. If this is the case, the smaller interval is merged with the biggest nearest interval to create a new larger fragment. If the new

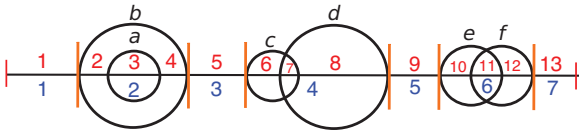


Figure 4: Fragment aggregation

interval satisfies the threshold constraint, it is confirmed as a fragment and the algorithm processes the next pair of interval boundaries. This process leads to a set of fragments, $F = \{f_1, ..., f_l\}$, where each fragment f_i has a corresponding scope $(t_{s,i}, t_{e,i})$ such that

(1) the length of each fragment is greater than or equal to the length lowerbound τ; i.e., $t_{e,i} - t_{s,i} \geq \tau$;
(2) the number of the fragments is suitably bounded; i.e., $l \leq \frac{n}{\tau}$; and
(3) the first and the last fragment boundaries are the first and the last point of the series, respectively; i.e., $x_{s,1} = 1$ and $x_{e,l} = n$.

Figure 4 presents an example: Here, the black circles represent the temporal scopes of the features. As we can see, using these features, we can potentially create up to 13 candidate fragments. Ideally, however, we would seek to avoid overly small fragments and therefore combine nearby intervals to generate 7 larger and more coherent fragments.

3.3 Fragment Characterization

Let F be the set of fragments of a time series, identified as above. The next step analyzes each fragment to extract characteristics to help compute the amount of smoothing that will be applied during the final filtering phase. Naturally, the degree of smoothing of a fragment will be a function of its local content and its relationship relative to the rest of the series. In this section, we consider two criteria:

- *entropy*: the information content of a fragment, measured using entropy, may indicate how much a fragment should be smoothed;
- *representativeness:* the similarity of a given fragment to the rest of the series may indicate how representative the fragment is and, therefore, may need to be considered during smoothing.

3.3.1 Entropy. We compute the entropy, $E(f_i)$ of a given fragment f_i by first quantizing the fragment into a quantization alphabet and counting the number of occurrences of each symbol:

Definition 3.1 (entropy). Let A be the quantization alphabet and pr_{ij} denotes the portion of times the symbol $a_j in A$ occurs in fragment f_i. The entropy [13], for the fragment f_i is defined as

$$E(f_i) = -\sum_{a_j \in A} pr_{ij} log(pr_{ij}).$$

3.3.2 Representativeness. In addition to entropy and feature scale, we also consider the similarity of a given fragment to the rest of the time series as a potential indicator of how representative the fragment is and, therefore, how much smoothing should be applied to the fragment. To compute the *representativeness* of the fragment $f_i \in F$, we first compute pairwise Dynamic Time Warping distances [12], $\Delta_{dtw}(f_i, f_j)$, between all pairs, f_i, f_j, of fragments in the series. We then create a similarity matrix Π, where the entry $\Pi[i, j]$ represents the similarity of fragment f_i to fragment f_j, defined as

$$\Pi[i, j] = 1 - \frac{\Delta_{dtw}(f_i, f_j)}{max_{f_h \in F}\{\Delta_{dtw}(f_i, f_h)\}}.$$

We then row-normalize the similarity matrix Π to obtain

$$\Pi'[i, j] = \frac{\Pi[i, j]}{\sum_{f_h \in F} \Pi[i, h]}.$$

Note that the transpose, $\mathcal{T} = \Pi'^T$, of the row-normalized matrix Π' can be considered as a random walk transition graph, where each column j corresponds to a fragment, f_j and the entry $\mathcal{T}[i, j]$ indicates the probability of random walk from fragment f_j to fragment f_i, such that the more similar the fragment f_i is to the fragment f_j the higher the probability of the random walk from f_j to f_i.

Given this transition matrix, $\mathcal{T}$, representing mutual similarities among the fragments, *CA-Smooth* then applies the well-known PageRank algorithm [3] to associate a degree of *representativeness* to each fragment in the graph:

Definition 3.2 (Representativeness). Let F be the set of all fragments and let $\mathcal{T}$ be the transition graph obtained as described above. Then, for $f_i \in F$, we compute its degree of representativeness, $R(f_i)$, as

$$R(f_i) = PR(f_i, \mathcal{T}),$$

where $PR(f_i, \mathcal{T})$ is the PageRank of fragment f_i based on the transition graph $\mathcal{T}$ that captures fragment similarities.

3.3.3 Characteristics of a Fragment. Given the above, we associate to each fragment $f_i \in F$ a combined characteristic, $C(f_i)$, defined as

$$C(f_i) = \langle E(f_i), R(f_i), \bar{E}(f_i), \bar{R}(f_i) \rangle,$$

where $\bar{E}$ and $\bar{R}$ are the inverses of E and R respectively. In this paper, the inverse $\bar{E}(f_i)$ is defined as

$$\bar{E}(f_i) = \frac{max_j(E(f_j)) - E(f_i)}{max_j(E(f_j)) - min_j(E(f_j))}$$

if $max_j(E(f_j)) \neq min_j(E(f_j))$; and $\bar{E}(f_i) = E(f_i)$ otherwise. Similarly for $\bar{R}$.

3.4 Content-Adaptive Filtering

Given a fragment f_i, *CA-Smooth* associates an overall smoothing scale to that fragment taking into account its characteristics, $C(f_i)$. More specifically, the smoothing scale $\sigma(f_i)$ corresponding to the fragment f_i is computed as

$$\forall_{f_i \in F} \ \ \sigma(f_i) = \Theta(C(f_i)),$$

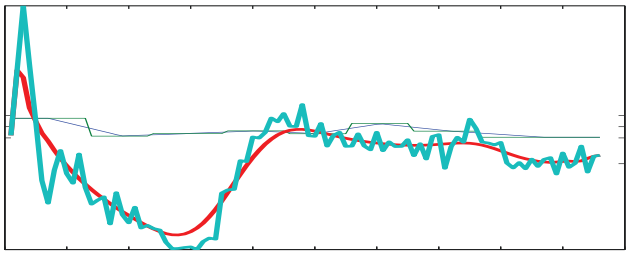


Figure 5: Adaptive smoothing of the series in Figure 2 revisited: here the thin green line indicates the degree of smoothing for different fragments of the time series; as we see in the figure, this is subject to abrupt changes – *CA-Smooth*, instead, adjusts the degree of smoothing on a per-instant basis (thin blue line)

where *CA-Smooth* uses the function Θ to adjust the degree of smoothing local to each fragment taking into account the fragments characteristics.[1].

3.4.1 Computing Instantaneous Smoothing Degrees. Note that, even though $\sigma(f_i)$ captures the local characteristics of f_i, applying this smoothing scale to each and every instant within the fragment would cause a significant problem at the fragment boundaries: since the amount of smoothing might change abruptly at the fragment boundaries, this can potentially introduce undesirable artifacts in the smoothed series. Therefore, *CA-Smooth* does not directly apply $\sigma(f_i)$ smoothing to each and every instant within the fragment, f_i. Instead, the algorithm varies the smoothing scale on a per-time-instant basis rather than on a per-fragment basis: further away from the fragment boundaries, the smoothing scale is governed by $C(f_i)$, but closer to the fragment boundaries (i.e., boundaries with f_{i-1} and f_{i+1}), the smoothing scale would be a combination of the smoothing scales of nearby fragments, eliminating abrupt changes on the smoothing filter (Figure 5). Intuitively, the smoothing scales themselves are smoothed to avoid smoothing artifacts at the fragment boundaries:

Let $F = \{f_1, \ldots, f_l\}$ be the set of fragments identified on the time series, T, $f_i \in F$ be a fragment, and let t_h be a time instance in f_i. Let also:

- $w_i = t_{e,i} - t_{s,i}$ denote the length of the frame f_i,
- $t_{h\perp} = t_h - w_{i-1}/2$,
- $t_{h\top} = t_h + w_{i+1}/2$,
- $f(t)$ denote a function that returns the fragment that contains the time instance t.

Given these, *CA-Smooth* computes the instantaneous smoothing scale, $\sigma(t_h)$, to be applied at time instant t_h as follows:

- if $f(t_{h\perp}) = f(t_{h\top}) = f_i$, then the time instance t_h is sufficiently away from the two boundaries of the fragment and we can apply the smoothing function $\sigma(t_h) = \sigma(f_i)$;
- if $t_{h\perp} < 1$ or $t_{h\top} > l$, then t_h is closer to the start or end of the time series than any other neighboring fragments and, thus, we again have $\sigma(t_h) = \sigma(f_i)$;
- if, however, $f(t_{h\perp}) = f_{i-1}$, then the point is closer to fragment f_{i-1} and, therefore, the smoothing scale will be a weighted average of the smoothing scales of the fragments f_{i-1} and f_i:

$$\sigma(t_h) = \frac{t_{s,i} - t_{h\perp}}{w}\sigma(f_{i-1}) + \frac{t_{h\top} - t_{s,i}}{w}\sigma(f_i);$$

- similarly, if $f(t_{h\top}) = f_{i+1}$, then the point is closer to fragment f_{i+1} and the smoothing scale will be a weighted average of the smoothing scales of the fragments f_i and f_{i+1}:

$$\sigma(t_h) = \frac{t_{h\top} - t_{e,i}}{w}\sigma(f_{i+1}) + \frac{t_{e,i} - t_{h\perp}}{w}\sigma(f_i).$$

Intuitively, we are fitting a length-w window around each time point t_h, where w corresponds to the length of the fragment f_i that contains the time point t_h. Given this window, the degree of smoothing at time t_h is computed based on how much this window overlaps[2] with the frames that come before or after the fragment f_i.

3.4.2 Adjusting Instantaneous Smoothing. Note that, so far, we have not considered the degree of smoothing, σ_g, global for the entire series. Intuitively, given a target degree, σ_g, for the series, the average of all instantaneous smoothing degrees should be σ_g. Therefore, in the final step, we adjust instantaneous smoothing degrees to reflect σ_g:

$$\sigma'(t_h) = \sigma_g \left(\frac{\sigma(t_h)}{\sum_{1 \le j \le n} \sigma(t_j)} \right).$$

Once the scale $\sigma(t_h)$ is computed for time instance, t_h, we apply a Gaussian filter with variance $\sigma(t_h)$, centered around t_h. As we discussed earlier, this corresponds to a filtering window $(t_h - 3\sigma(t_h), t_h + 3\sigma(t_h))$. At the very boundaries of the time series, where the start or end of the smoothing interval may fall beyond the boundaries, we suitably scale $\sigma(t_h)$ to ensure that smoothing filter always falls within the time series.

4 EVALUATION

In this section, we present case studies to illustrate the impact of different content-adaptive smoothing criteria. For this purpose, we use six datasets 1 with different temporal characteristics. These data sets, listed in Table 1, are publicly available at [1].

[1]We show the impact of different Θ functions in the experimental evaluation section.

[2]Note that the window can overlap with only one of the fragments f_{i-1} or f_{i+1} at a time, never both; note also that the only situation in which the window does not overlap with either of the two neighboring fragments is when t_h is at the center of the fragment f_i.

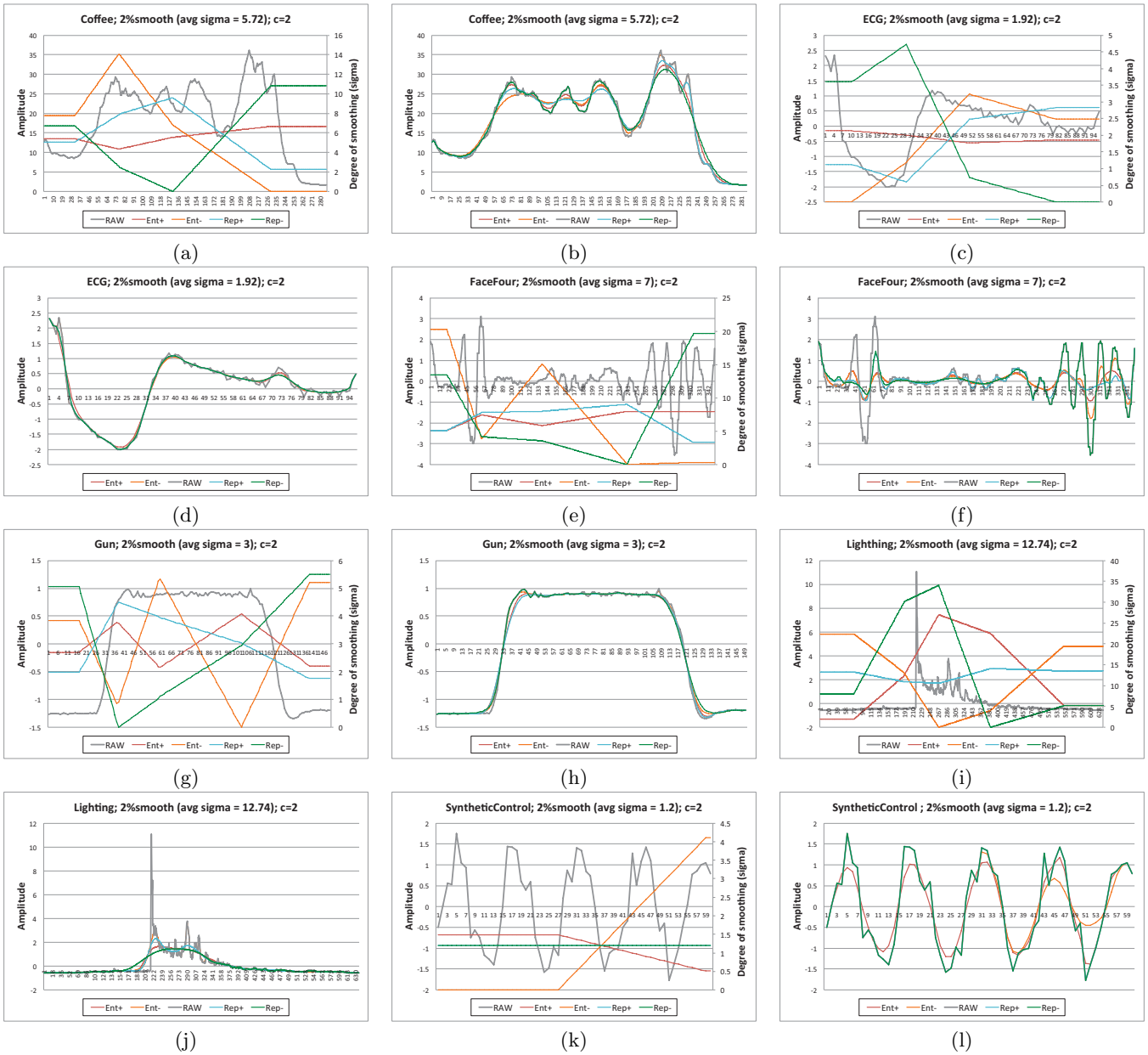


Figure 6: Impact of different filtering strategies

4.1 Configurations

In Table 2, we report the configuration parameters we consider. Here, σ_g is the global smoothing parameter, which would be used by a fixed smoothing strategy. The adaptive methods considered in this paper adapt the instantaneous degree of smoothing based on the characteristics of the data, in such a way that the overall average smoothing across the entire series is equal to σ_g.

As described in Section 3.2, the parameter τ is the length of the smallest fragment created and is a multiple, c, of the of the average smoothing window size, $(6 \times \sigma_g)$. In this paper, we rely on the process described in [4] for extracting locally-salient robust features (Section 3.1) which are then used for identifying the fragment boundaries (Section 3.2); σ_0 and O are two parameters used by this algorithm to control the feature sizes. In particular, the size of the smallest feature identified by the algorithm is $6 \times \sigma_0$, whereas the size of the largest feature is $6 \times \sigma_0 \times 2^O$. As we see in Table 2, the sizes of these features are also a function of the target smoothing rate, σ_g.

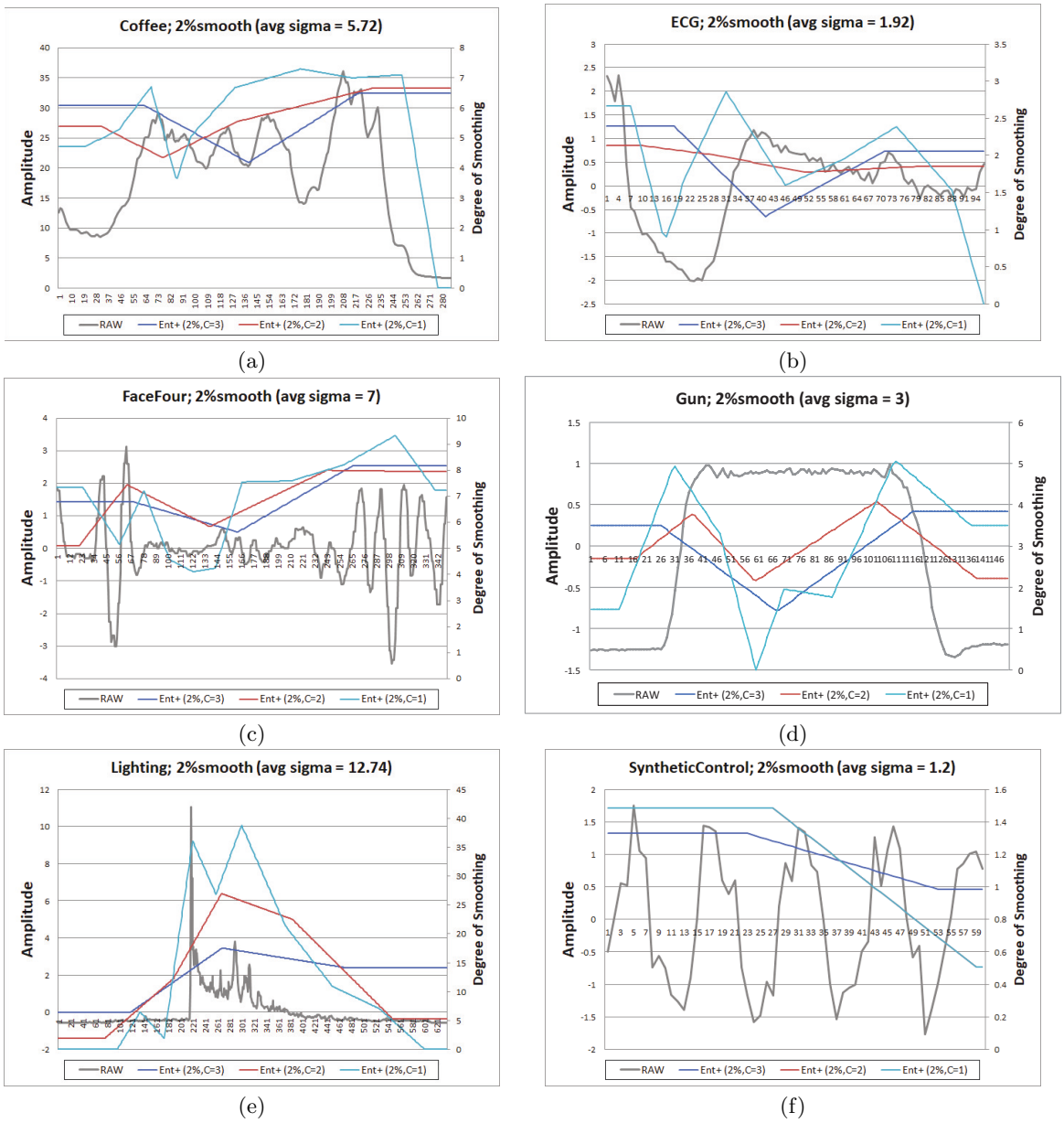


Figure 7: Impact of different fragment length thresholds

Table 1: Datasets Characteristics

DataSet	#series
Coffee2	286
FaceFour	350
Gun	150
ECG200	96
synthetic_control	60
Lighting2_TEST	638

Table 2: Experiments Configurations

Parameter	Value
σ_g	2% of the series length
τ	$c \times (6 \times \sigma_g)$
c	1,2,3
σ_0	$\frac{c}{2} * \sigma_g$
# of octaves (O)	2

4.2 Results

In Figure 6 we present the smoothing samples for the six data sets; in particular, figures labeled (a), (c), (e),(g),(i), and (k) show the instantaneous smoothing degrees computed by *CA-Smooth* under four entropy and representativeness based smoothing criteria discussed in Section 3.3.3, whereas figures labeled (b), (d), (f), (h), (j), and (l) show the resulting content-adapted smoothed series.

Let us consider the Gun series presented in Figures 6(g) and 6(h). As we see in Figures 6(g), the entropy based criteria E (Ent+ in the figure) is active where there is major change in the series and, as expected, the opposite criteria, $\bar{E}$ (Ent- in the figure), provides higher instantaneous smoothing weights where the series are relatively flat. The figure also shows that different regions of the series are associated with different degrees of representativeness as described in Section 3.3.2. In the figure *Rep+* corresponds to the criterion R which applies high smoothing on parts of the series with high representativeness, whereas *Rep−* corresponds to the opposite criterion, $\bar{R}$, which smooths primarily parts of the series that are not representative. As we see in Figures 6(h), different criteria lead to different degrees of smoothing especially at the parts of the series with large change – the appropriate criteria to select would be a function of the underlying task.

Finally, Figure 7 shows the impact of different values of c, which controls the lowerbound, τ, of fragment sizes, on the instantaneous smoothing. As we see in these figures, for lower values of c, we obtain more fragments and the instantaneous smoothing degrees becomes more impacted by the very local characteristics of the series; on the other hand, as the value of the c increases, the number of fragments reduces and the degree of instantaneous smoothing becomes impacted by the characteristics of larger fragments.

5 CONCLUSIONS

Arguing that time series smoothing must be performed in a way that is adaptive to the temporally-varying characteristics of the input series, in this paper, we proposed a content-adaptive smoothing strategy, *CA-Smooth*. The proposed technique relies on a locally-robust feature extraction approach to locate robust time series fragments and then identify characteristics, such as entropy and representativeness, of these fragments. *CA-Smooth* treats different parts of the time series according to the discovered fragment characteristics and associates instantaneous smoothing weights to each time instant in a way that represents both the user's global smoothing target and the local series characteristics.

REFERENCES

[1] Time series dataset at UC Riverside. https://www.cs.ucr.edu/~eamonn/time_series_data/.

[2] An evaluation of time-series smoothing algorithms for land-cover classifications using modis-ndvi multi-temporal data. *Remote Sensing of Environment*, 174:258 – 265, 2016.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[4] K. S. Candan, R. Rossini, X. Wang, and M. L. Sapino. sdtw: computing dtw distances using locally relevant constraints based on salient feature alignments. *Proceedings of the VLDB Endowment*, 5(11):1519–1530, 2012.

[5] X. Chen, V. Mithal, S. R. Vangala, I. Brugere, S. Boriah, and V. Kumar. A study of time series noise reduction techniques in the context of land cover change detection. *Department of Computer Science and Engineering-Technical Reports, no. TR*, pages 11–016, 2011.

[6] J. Gao, H. Sultan, J. Hu, and W.-W. Tung. Denoising nonlinear time series by adaptive filtering and wavelet shrinkage: a comparison. *IEEE signal processing letters*, 17(3):237–240, 2009.

[7] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In *In an Edited Volume, Data mining in Time Series Databases. Published by World Scientific*, pages 1–22. Publishing Company, 1993.

[8] J. Large, P. Southam, and A. Bagnall. Can automated smoothing significantly improve benchmark time series classification algorithms? In H. Pérez García, L. Sánchez González, M. Castejón Limas, H. Quintián Pardo, and E. Corchado Rodríguez, editors, *Hybrid Artificial Intelligent Systems*, pages 50–60, Cham, 2019. Springer International Publishing.

[9] W.-H. Lee, J. Ortiz, B. Ko, and R. Lee. Time series segmentation through automatic feature learning, 2018.

[10] Y. Qi and K. S. Candan. CUTS: *CU*rvature-based development pattern analysis and segmentation for blogs and other *T*ext *S*treams. In *HYPERTEXT 2006, Proceedings of the 17th ACM Conference on Hypertext and Hypermedia, August 22-25, 2006, Odense, Denmark*, pages 1–10, 2006.

[11] K. Rong and P. Bailis. ASAP: prioritizing attention via time series smoothing. *PVLDB*, 10(11):1358–1369, 2017.

[12] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[13] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[14] K. Shin, J. Hammond, and P. White. Iterative svd method for noise reduction of low-dimensional chaotic time series. *Mechanical Systems and Signal Processing*, 13(1):115–124, 1999.

[15] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor. Detecting trend and seasonal changes in satellite image time series. *Remote sensing of Environment*, 114(1):106–115, 2010.

[16] S. Winsberg. Jeffrey s. simonoff. smoothing methods in statistics. *PSYCHOMETRIKA*, 62:163–164, 1997.