Adaptive, Hands-Off Stream Mining

Spiros Papadimitriou* Anthony Brockwell† Christos Faloutsos*

Computer Science Department* and
Department of Statistics†
Carnegie Mellon University,
Pittsburgh, PA, USA
{spapadim,christos}@cs.cmu.edu, abrock@stat.cmu.edu

Abstract

Sensor devices and embedded processors are becoming ubiquitous. Their limited resources (CPU, memory and/or communication bandwidth and power) pose some interesting challenges. We need both powerful and concise "languages" to represent the important features of the data, which can (a) adapt and handle *arbitrary* periodic components, including bursts, and (b) require little memory and a single pass over the data.

We propose AWSOM (Arbitrary Window Stream mOdeling Method), which allows sensors in remote or hostile environments to efficiently and effectively discover interesting patterns and trends. This can be done *automatically*, i.e., with no user intervention and expert tuning before or during data gathering. Our algorithms require limited resources and can thus be incorporated in sensors, possibly alongside a distributed query processing engine [9, 5, 22]. Updates are performed in constant time, using logarithmic space. Existing, state of the art forecasting methods (SARIMA, GARCH, etc) fall short on one or more of these requirements. To the best of our knowledge, AWSOM is the first method that has all the above characteristics.

Experiments on real and synthetic datasets demonstrate that AWSOM discovers meaningful patterns over long time periods. Thus, the patterns can also be used to make long-range forecasts, which are notoriously difficult to perform. In fact, AWSOM outperforms manually set up auto-regressive models, both in terms of long-term pattern detection and modeling, as well as by at least 10× in resource consumption.

1 Introduction

Several applications produce huge amounts of data in the form of a semi-infinite stream of values [17, 15, 12, 14], typically samples or measurements. Formally, a stream is a discrete sequence of numbers $X_0, X_1, \ldots, X_t, \ldots$. Time sequences have attracted attention [6], for forecasting in financial, sales, environmental, ecological and biological time series, to mention a few. However, several new and exciting applications have recently become possible.

The emergence of cheap and small sensors has attracted significant attention. Sensors are small devices that gather measurements—for example, temperature readings, road traffic data, geological and astronomical observations, patient physiological data, etc. There are numerous, fascinating applications for such sensors and sensor networks, in fields such as health care and monitoring, industrial process control, civil infrastructure [8], road traffic safety and smart houses, to mention a few. Although current small sensor prototypes [19] have limited resources (512 bytes to 128Kb of storage), dime-sized devices with memory and processing power equivalent to a PDA are not far away.

†This material is based upon work supported by the National Science Foundation under Grants No. DMS-9819950 and IIS-0083148.

*This material is based upon work supported by the National Science Foundation under Grants No. IIS-9817496, IIS-9988876, IIS-0083148, IIS-0113089, IIS-0209107 IIS-0205224 by the Pennsylvania Infrastructure Technology Alliance (PITA) Grant No. 22-901-0001, and by the Defense Advanced Research Projects Agency under Contract No. N66001-00-1-8936. Additional funding was provided by donations from Intel. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, DARPA, or other funding parties.

Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003

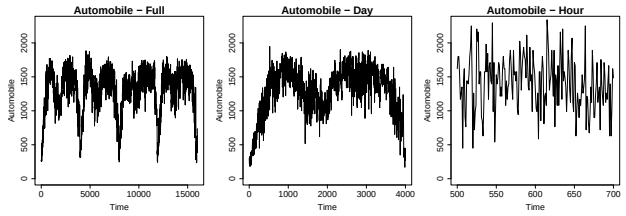


Figure 1: *Automobile traffic, complete series, one day and one hour (the first two are 8- and 4-point averages to make the trends easier to see). There is clearly a daily periodicity. Also, in each day there is another distinct pattern (morning and afternoon rush hours). However, at an hour scale traffic is highly bursty—in fact, it can be modeled by* self-similar *noise. We want a method that can capture all this information automatically, with one pass and using limited memory!*

In fact, PDA-like devices with data gathering units are already being employed in some of the above applications. The goal in the next decade is single-chip computers with powerful processors and 2–10Gb [8] of nonvolatile storage. Furthermore, embedded processors are becoming ubiquitous and their power has yet to be harnessed. A few examples of such applications are (a) intelligent (*active*) disks [24] that learn input traffic patterns and do appropriate prefetching and buffering, (b) intelligent routers that monitor data traffic and simplify network management.

From now on, we use the term "sensor" broadly, to refer to any embedded computing device with fairly limited processing, memory and (optionally) communication resources and which generates a semi-infinite sequence of measurements.

The resource limitations unavoidably imply the need for certain trade-offs—it is impossible to store everything. Furthermore, we want to make the most of available resources, allowing the sensor to adapt and operate without supervision for as long as possible. This is the problem we address in this work. The goal is a "language" (i.e., model/representation) for efficient and effective, automatic stream mining. We want to collect information, in real-time and without any human intervention, and discover patterns such as "the hourly phone call volume so far follows a daily and a weekly periodicity, with bursts roughly every year" (which a human might recognize as, e.g., the Mother's day surge).

This problem is orthogonal to that of continuous query processing. We focus on an adaptive algorithm that can look for arbitrary patterns and requires no *initial* human intervention to guide it. There are situations when we do not know *beforehand* what we are looking for. Furthermore, it may be impossible to guide the sensor as it collects data, due to the large volume of data and/or limited or unavailable communication. If further exploration is desired, users can issue further queries, guided by the general long-term patterns to quickly narrow down the "search space."

In detail, the main requirements are (see also Figure 1): **(1) No human in the loop:** In a general sensor setting we *cannot afford* human intervention. **(2) Periodic component identification:** Humans can achieve this task, visually, from the time-plot. Our method should automatically spot multiple periodic components, each of unknown, *arbitrary* period. **(3) Online, one-pass algorithm:** We can afford neither the memory or time for offline updates, much less multiple passes over the data stream. **(4) Limited memory:** Sensor memory will be exhausted, unless our method carefully detects redundancies (or equivalently, patterns) and exploits them. Furthermore, we want to collect data even when network connectivity is intermittent (e.g., due to power constraints) or even non-existent. **(5) Simple, but powerful patterns:** We need simple patterns (i.e., equations, rules) which can be easily communicated to other nearby sensors and/or to a central processing site. These patterns should be powerful enough to capture most of the regularities in real-world signals. **(6) Any-time forecasting/outlier detection:** It is not enough to do compression (e.g., of long silence periods, or by ignoring small Fourier or wavelet coefficients). The model should be *generative* and thus able report outliers. An outlier can be defined as any value that deviates too much from our forecast (e.g., by two standard deviations). AWSOM has all of these characteristics, while none of the previously published methods (AR and variations, Fourier analysis, wavelet decomposition—see Section 2.2) can claim the same.

2 Related work

Previous work broadly falls into two categories. The first includes work done by the databases community on continuous query processing. These methods employ increasingly sophisticated mathematical methods, but the focus is typically on some form of compression (or, *synopses*) which do not employ generative models. The other includes various statistical methods for time series forecasting. However, these require the intervention of trained statisticians and are computationally intensive.

Thus the authors believe that there is a need for straightforward methods of time series model building which can be applied in real-time to semi-infinite streams of data, using limited memory.

2.1 Continuous queries and stream processing

An interesting method for discovering *representative trends* in time series using *sketches* was proposed by Indyk et. al. [20]. A representative trend is a section of the time series that has the smallest sum of "distances" from *all* other sections of the same length. The proposed method employs random projections for dimensionality reduction and FFT to quickly compute the sum of distances. However, it cannot be applied

to semi-infinite streams, since each section has to be compared to every other.

Gilbert et. al. [17] use wavelets to "compress" the data into a fixed amount of memory, by keeping track of the largest Haar wavelet coefficients and carefully updating them on-line (in the following, we will use the name *Incremental DWT* or *IncDWT* for short). However, this method does not try to discover patterns and trends in the data. Thus, it cannot compete directly with our method, which employs a *generative* model. More recently, Garofalakis et. al. [14] presented an approach for accurate data compression using *probabilistic wavelet synopses*. However, this method has an entirely different focus and cannot be applied to semi-infinite streams. The recent work of Zhang et. al. [18] efficiently constructs ϵ-accurate histograms in a stream setting, but these synopsize data over a fixed time window. Further work on streams focuses on providing exact answers to pre-specified sets of queries using the minimum amount of memory possible. Arvind et. al. [2] study the memory requirements of continuous queries over *relational* data streams. Datar et. al. [11] keep *exact* summary statistics and provide theoretical bounds in the setting of a bit stream.

There is also recent work on approximate answers to various types of continuous queries. Gehrke et. al. [15] presents a comprehensive approach for answering correlated aggregate queries (e.g., "find points below the (current) average"), using histogram "summaries" to approximate aggregates. Dobra et. al. [12] present a method for approximate answers to aggregate multi-join queries over several streams, using random projections and boosting.

Zhang et. al. [28] present a framework for spatio-temporal joins using multiple-granularity indices. Aggregation levels are pre-specified and the main focus is on efficient indexing. Also, [26] present a method for analysis of multiple co-evolving sequences. A system for linear pattern discovery on multi-dimensional time series was presented recently in [10]. Although this framework employs varying resolutions *in time*, it does so by straight aggregation, using manually selected aggregation levels (although the authors discuss the use of a geometric progression of time frames) and can only deal with, essentially, linear trends. Finally, very recently, Bulut et. al. [7] proposed an approach for hierarchical stream summarization (similar to that of [17]) which focuses on simple queries and communication/caching issues for wavelet coefficients.

2.2 Time series methods

None of the continuous querying methods deal with pattern discovery and forecasting. The typical approaches to forecasting (i.e., generative time series models) include the traditional *auto-regressive (AR)* models and their generalizations, *auto-regressive moving average (ARMA)*, *auto-regressive integrated moving average (ARIMA)* and *seasonal ARIMA (SARIMA)* [6]. Although popular, these methods fail to meet many of the requirements listed in the introduction. The most important failure is that they need human intervention and fine-tuning. As mentioned in statistics textbooks such as [6]:

> *"The first step in the analysis of any time series is to plot the data. [...] Inspection of a graph may also suggest the possibility of representing the data as a realization of [the 'classical decomposition' model]."*

Thus, such methods are not suited for remote, unsupervised operation. Furthermore, these methods have a number of other limitations.

Existing model-fitting methods are typically batch-based (i.e., do not allow online update of parameters). Established methods for determining model structure are at best computationally intensive, besides not easily automated.

If there are non-sinusoidal periodic components, ARIMA models will miss them completely. Large window sizes introduce severe estimation problems, both in terms of resource requirements as well as accuracy.

In addition, ARIMA models cannot handle bursty time series, even when the bursts are re-occurring. While *GARCH (generalized auto-regressive conditional heteroskedasticity)* models [4] can handle the class of bursty *white noise* sequences, the computational difficulties involved are prohibitive. Recently, the ARIMA model has been extended to *ARFIMA (auto-regressive fractionally integrated moving average)*, which handles the class of *self-similar* bursty sequences [3]. However, ARFIMA is even harder to use than ARIMA.

All the above methods deal with linear forecasting. Non-linear modeling methods [25] also require human intervention to choose the appropriate windows for non-linear regression or to configure an artificial neural network.

2.3 Other

There is a large body of work in the signal processing literature related to compression and feature extraction. Typical tools include the Fast Fourier Transform (FFT), as well as the Discrete Wavelet Transform (DWT) [23]. However, most of the algorithms (a) deal with *fixed length* signals of size N, and (b) cannot do forecasting (i.e., do not employ a *generative* model).

3 Background material

In this section we give a very brief introduction to some necessary background material.

3.1 Auto-regressive modeling

Auto-regressive models are the most widely known and used. Due to space constraints, we present the basic

Symbol	Definition
X_t	Value at time $t = 0, 1, \ldots$.
N	Number of points *so far* from $\{X_t\}$.
$W_{l,t}$	Wavelet coefficient (level l, time t).
$V_{l,t}$	Scaling coefficient (level l, time t).
$\beta_{\delta l,\delta t}$	AWSOM coefficient, $(\delta l, \delta t) \in \mathcal{D}$).
$\mathcal{D}$	Set of window offsets for AWSOM.
AWSOM$(n_0, \ldots, n_\lambda)$	Offsets per level $(n_0, \ldots, n_\lambda)$ in $\mathcal{D}$—see Definition 5. $(n_0, \ldots, n_\lambda)$ is also called the AWSOM *order*.
λ	Depth of AWSOM model, $\lambda \geq 0$.
k	Total order of an AWSOM model $k \equiv \lvert\mathcal{D}\rvert = \sum_{l=0}^{\lambda} n_l$.

Table 1: *Symbols and definitions.*

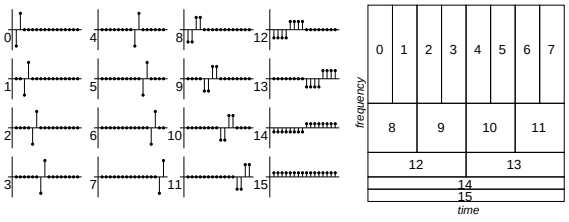


Figure 2: *Haar bases and correspondence to time/frequency (for signal length $N = 16$). Each wavelet coefficient is a linear projection of the signal to the respective basis.*

ideas—more information can be found in, e.g., [6]. The main idea is to express X_t as a function of its previous values, plus (filtered) noise ϵ_t:

$$X_t = \phi_1 X_{t-1} + \ldots + \phi_W X_{t-W} + \epsilon_t \quad (1)$$

where W is a window that is determined by trial and error, or by using a criterion that penalizes model complexity (i.e., large values of W), like the *Akaike Information Criterion (AIC)*. Seasonal variants (SAR, SAR(I)MA) also use window offsets that are multiples of a single, fixed period (i.e., besides terms of the form X_{t-i}, the equation contains terms of the form X_{t-Si} where S is a constant). AR(I)MA requires preprocessing by trained statisticians to remove trends and seasonalities, typically by visual inspection of the sequence itself, as well as its *Auto-Correlation Function (ACF)*.

Recursive least squares. *Recursive Least Squares* (*RLS*) is a method that allows dynamic update of a least-squares fit. More details are given in Appendix B and in [27].

3.2 Wavelets

The N-point *discrete wavelet transform (DWT)* of a length N time sequence gives N wavelet coefficients. Wavelets are best introduced with the Haar transform, because of its simplicity (a more rigorous introduction can be found, e.g., in [23]). At each level l of the construction we keep track of two sets of coefficients, each of which "looks" at a time window of size 2^l:

- $W_{l,t}$: The *smooth* component, which consists of the $N/2^l$ *scaling coefficients*. These capture the low-frequency component of the signal; in particular, the frequency range $[0, 1/2^l]$.
- $V_{l,t}$: The *detail* component, which consists of the $N/2^l$ *wavelet coefficients*. These capture the high-frequency component; in particular, the range $[1/2^l, 1/2^{l-1}]$.

The construction starts with $V_{0,t} = X_t$ and $W_{0,t}$ is not defined. At each iteration $l = 1, 2, \ldots, \lg N$ we perform two operations on $V_{l-1,t}$ to compute the coefficients at the next level:

- Differencing, to extract the high frequencies:
$$W_{l,t} = (V_{l-1,2t} - V_{l-1,2t-1})/\sqrt{2}$$
- Smoothing, which averages[1] each consecutive pair of values and extracts the low frequencies:
$$V_{l,t} = (V_{l-1,2t} + V_{l-1,2t-1})/\sqrt{2}$$

We stop when $W_{l,t}$ consists of one coefficient (which happens at $l = \lg N + 1$). The scaling coefficients are needed only during the intermediate stages of the computation. The final wavelet transform is the set of all wavelet coefficients along with $V_{\lg N+1,0}$. Starting with $V_{\lg N+1,0}$ (which is also referred to as the signal's scaling coefficient) and following the inverse steps, we can reconstruct each $V_{l,t}$ until we reach $V_{0,t} \equiv X_t$.

Figure 2 illustrates the final effect for a signal with $N = 16$ values. Each wavelet coefficient is the result of projecting the original signal onto the corresponding basis signal. Figure 2 shows the *scalogram*, that is, the energy (i.e., squared magnitude) of each wavelet coefficient versus the location in time and frequency it is "responsible" for. In general, there are many wavelet transforms, but they all follow the pattern above: a wavelet transform uses a pair of filters, one high-pass and one low-pass.

For our purposes here, we shall restrict ourselves to wavelets of the Daubechies family, which have desirable smoothness properties and successfully compress many real signals. In practice, although by far the most commonly used (largely due to their simplicity), Haar wavelets are too unsmooth and introduce significant artifacting [23]. In fact, unless otherwise specified, we use Daubechies-6.

Incremental wavelets. This part is a very brief overview of how to compute the DWT incrementally. This is the main idea of IncDWT [17], which uses Haar wavelets. In general, when using a wavelet filter of length L, the wavelet coefficient at a particular level is computed using the L corresponding scaling coefficients of the previous level. Recall that $L = 2$ for Haar (average and difference of two consecutive points), and $L = 6$ for Daubechies-6 that we typically use. Thus, we need to remember the last $L - 1$ scaling coefficients at each level. We call these the *wavelet crest*.

[1]The scaling factor of $1/\sqrt{2}$ in both the difference and averaging operations is present in order to preserve total signal energy (i.e., sum of squares of all values).

Definition 1 (Wavelet crest) *The* wavelet crest *at time t is defined as the set of scaling coefficients (wavelet smooths) that need to be kept in order to compute the new wavelet coefficients when X_t arrives.*

Lemma 1 (DWT update) *Updating the wavelet crest requires space $(L-1)\lg N + L = O(L \lg N) = O(\lg N)$, where L is the width of the wavelet filter (fixed) and N the number of values seen so far.*

Proof See [17]. Generalizing to non-Haar wavelets and taking into account the wavelet filter width is straightforward. ■

3.2.1 Wavelet properties

In this section we emphasize the DWT properties which are relevant to AWSOM.

Computational complexity. The DWT can be computed in $O(N)$ time and, as new points arrive, it can be updated in $O(1)$ amortized time. This is made possible by the structure of the time/frequency decomposition which is unique to wavelets. For instance, the Fourier transform also decomposes a signal into frequencies (i.e., sum of sines), but requires $O(N \lg N)$ time to compute and cannot be updated as new points arrive.

Time/frequency decomposition. Notice (see scalogram in Figure 2) that higher level coefficients are highly localized in time, but involve uncertainty in frequency and vice-versa. This is a *fundamental* trade-off of any time/frequency representation and is a manifestation of the *uncertainty principle*, according to which localization in frequencies is inversely proportional to localization in time. The wavelet representation is an excellent choice when dealing with semi-infinite streams in limited memory: it "compresses" well many real signals, while it is fast to compute and can be updated online.

Wavelets and decorrelation. A wavelet transform with length $2L$ filter can decorrelate only certain signals provided their L-th order (or less) backward difference is a stationary random process [23]. For real signals, this value of L is not known in advance and may be impractically large: the space complexity of computing new wavelet coefficients is $O(L \lg N)$—see Lemma 1.

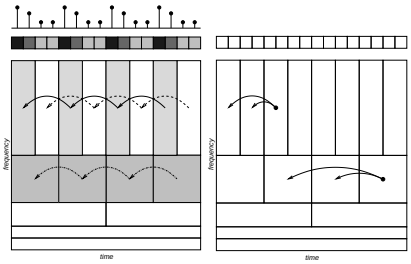
Wavelet variance. One further benefit of using wavelets is that they decompose the variance across scales. Furthermore, the plot of log-power versus scale can be used to detect self-similar components (see Appendix A for a brief overview).

4 Proposed method

In this section we introduce our proposed model. What equations should we be looking for to replace ARIMA's (see Equation 1)?

Method	Contin. Streams	Trends / Forecast	Automatic	Memory
DFT (N-point)	NO	NO	—	—
SWFT (N-point)	YES(?)	NO	—	—
DWT (N-point)	NO	NO	—	—
IncDWT [17]	YES	NO	—	—
Sketches [20]	NO	YES(?)	—	—
AR / ARIMA	YES	YES	NO [6]	W^2
AWSOM	YES	YES	YES	$m\lvert\mathcal{D}\rvert^2$

Table 2: *Comparison of methods.*



(a) (b)

Figure 3: AWSOM—*Intuition and demonstration.* AWSOM *captures intra-scale correlations (a). Also, (b) demonstrates why we fit different models per level.*

4.1 Intuition behind our method

First part—information representation. Traditional models (such as ARIMA) operate directly in the time domain. Thus, they cannot deal with redundancies, seasonalities, long-range behavior, etc. This is where a human expert is needed to manually detect these phenomena and transform the series to match ARIMA's assumptions.

This is a crucial choice—is there a better one? We want a powerful and flexible representation that can adapt to the sequence, rather than expect someone to adapt the sequence to the representation. We propose to use wavelets because they are extremely successful in compressing most real signals, such as voice and images [13], seismic data [29], biomedical signals [1] and economic time sequences [16]. By using wavelet coefficients, we immediately discard many redundancies (i.e., near-zero valued wavelet coefficients) and focus what really matters. Furthermore, the DWT can be computed quickly and updated online.

Second part—correlation. In the wavelet domain, how can we capture arbitrary periodicities? A periodic signal will have high-energy wavelet coefficients at the scales that correspond to its frequency. Also, successive coefficients on the same level should have related values (see Figure 3(a)). Thus, in order to capture periodic components, we should look for intra-scale correlations between wavelet coefficients.

How should we capture bursts? Short bursts carry energy in most frequencies. Therefore wavelet coefficients across scales will carry a large energy (see Figure 14(a)). If the phenomenon follows some pattern, then it is likely that there will be an inter-scale cor-

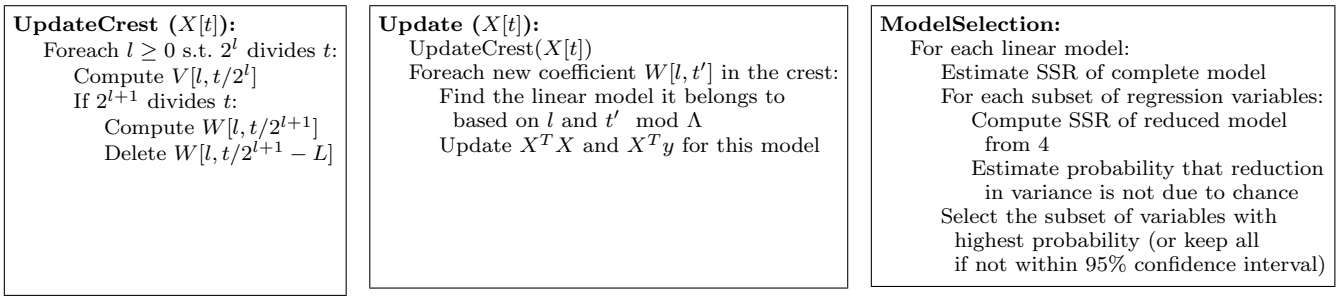


Figure 4: *High-level description of the algorithms.*

relation among several of the wavelet coefficients (see Appendix D for more details).

The last question we need to answer is: what type of regression models should we use to quantify these correlations? Our proposed method tries to capture inter- and intra-scale correlations by fitting a linear regression model in the wavelet domain. These can also be updated online with RLS.

To summarize, we propose using the wavelet representation of the series and capturing correlations in the wavelet domain (see Figure 3(b)).

4.2 AWSOM modeling

In the simplest case, we try to express the wavelet coefficients at each level as a function of the n_0 previous coefficients of the same level, i.e.,

$$W_{l,t} = \beta_{0,1} W_{l,t-1} + \beta_{0,2} W_{l,t-2} + \cdots + \beta_{0,n_0} W_{l,t-n_0}$$

where $W_{l,t}$ are the wavelet coefficients (at time t and level l) and $\beta_{0,i}^{(l)}$ are the AWSOM coefficients (for level l and lag i). We estimate one set of such coefficients for *each* level l. This is a model of order n_0, denoted as AWSOM(n_0).

This can capture arbitrary periodic components and is sufficient in many real signals. In general, besides within-scale correlations, we may also try to capture across-scale correlations by also including terms of the form $\beta_{\delta l,\delta t} W_{l-\delta l,t/2^{\delta l}-\delta t}$ in the above equation. When we use $n_{\delta l}$ coefficients from each level in the regression models, the order is $(n_0, n_1, \ldots, n_{\delta l})$ and the model is denoted by AWSOM($n_0, n_1, \ldots, n_{\delta l}$). See Appendix D for details.

4.3 Model selection

Many of the dependences may be statistically insignificant, so the respective coefficients should be set to zero. We want to (a) avoid over-fitting and (b) present to the user those patterns that are important. We can do model selection and combination—technical details are in Appendix C. This can be performed using *only* data gathered online and time complexity is *independent* of the stream size. The only thing that needs to be decided in advance is the largest AWSOM($n_0, \ldots, n_\lambda$) order we may want to fit. From the data collected, we can then automatically select any model of smaller order (AWSOM($n'_0, \ldots, n'_{\lambda'}$), where $\lambda' \leq \lambda$ and $n'_i \leq n_i$).

Dataset	Size	Description
`Triangle`	64K	Triangle wave (period 256)
`Mix`	256K	Square wave (period 256) plus sine (period 64)
`Sunspot`	2K	Sunspot data
`Automobile`	32K	Automobile traffic sensor trace from large Midwestern state

Table 3: *Description of datasets (sizes are in number of points, 1K=1024 points).*

4.4 Complexity

In this section we show that our proposed AWSOM models can be easily estimated with a single-pass, "any-time" algorithm. From Lemma 1, estimating the new wavelet coefficients requires space $O(\lg N)$. In fact, since we typically use Daubechies-6 wavelets ($L = 6$), we need to keep exactly $5 \lg N + 6$ values. The AWSOM models can be dynamically updated using RLS.

The number of active levels L_a depends on N and the value of T is fixed and depends only on the depth λ. In particular:

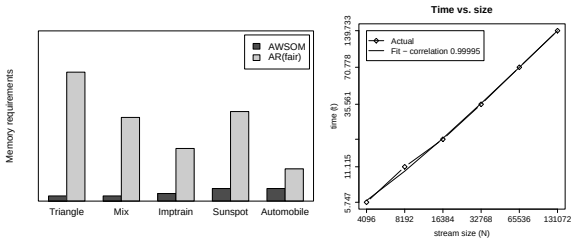
$$T \sim 2^\lambda \quad \text{and} \quad L_a \sim \lg \frac{N}{N_a T} = \lg N - \lg N_\Lambda - \lambda$$

This leads to the following result:

Lemma 2 (Logarithmic space complexity) *Maintaining the model requires $O(\lg N + mk^2)$ space, where N is the length of the signal so far, k is the total* AWSOM *order and $m = L_a T$ the number of equations.*

Proof Keeping the wavelet crest scaling coefficients requires space $O(\lg N)$. If we use recursive least squares, we need to maintain a $k \times k$ matrix for each of the m equations in the model. ■

Auto-regressive models with a comparable window size need space $O(m^2 k^2)$, since the equivalent fair window size is $W \approx mk$. Here, "fair" means that the number of total number of AWSOM coefficients plus the number of initial conditions we need to store is the same for both methods. This is the information that comprises the data synopsis and that would have



(a) (b)

Figure 5: *(a) Memory space requirements (normalized): Space needed to keep the models up-to-date (*AWSOM *and AR with equivalent, fair window size. (b) Time complexity versus stream size (Python prototype), including model selection; the relationship is exactly linear, as expected.*

to be eventually communicated. However, the device gathering the measurements needs extra storage space in order to update the models. The latter is, in fact, *much* larger for AR than for AWSOM (see Figure 5(a)). Thus this definition of equivalent window actually favors AR.

Lemma 3 (Time complexity) *Updating the model when a new data point arrives requires $O(k^2)$ time on average, where k is the number of* AWSOM *coefficients in each equation.*

Proof On average, the wavelet crest scaling coefficients can be updated in $O(1)$ amortized time. Although a single step may require $O(\lg N)$ time in the worst case, on average, the (amortized) time required is $O(\sum_{i=0}^{n} \mathcal{B}(i)/N) = O(1)$ (where $\mathcal{B}(i)$ is the number of trailing zeros in the binary representation of i)[2]. Updating the $k \times k$ matrix for the appropriate linear equation (which can be identified in O(1), based on level l and on $t \mod T$), requires time $O(k^2)$. ■

Auto-regressive models with a comparable window size need $O(m^2k^2)$ time per update.

Corollary 1 (Constant-time update) *When the model parameters have been fixed (typically k is a small constant ≈ 10 and $m \sim \lg N$), the model requires space $O(\lg N)$ and amortized time $O(1)$ for each update.*

Figure 5(b) shows that this is clearly the case, as expected.

5 Experimental evaluation

We compared AWSOM against standard AR (with the equivalent, fair window size—see Section 4.4), as well as hand-tuned (S)ARIMA (wherever possible). Our prototype AWSOM implementation is written in Python, using Numeric Python for fast array manipulation. We used the standard `ts` package from R

[2]Seen differently, *IncDWT* is essentially an pre-order traversal of the wavelet coefficient tree.

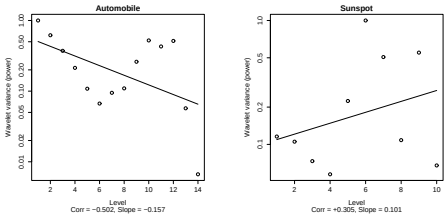


Figure 6: *Wavelet variance diagnostic.* `Automobile` *exhibits self-similarity in scales up to 6 (which roughly corresponds to one hour) but not overall.*

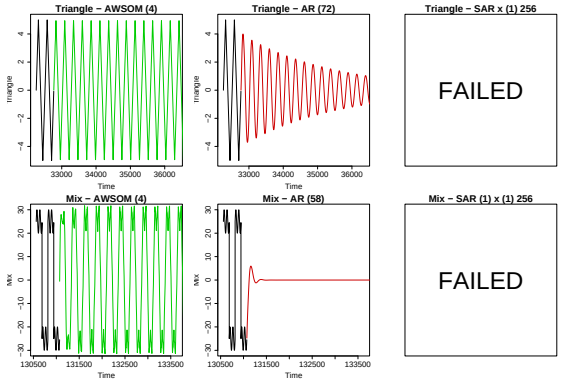


Figure 7: *Forecasts—synthetic datasets. Note that AR gives the wrong trend (if any), while seasonal AR fails to complete.*

(version 1.6.0—see `http://www.r-project.org/`) for AR and (S)ARIMA models. We illustrate the properties of AWSOM and how to interpret the models using synthetic datasets and then show how these apply to real datasets (see Table 3).

Only the first half of each sequence was used to estimate the models, which were then applied to generate a sequence of length equal to that of the *entire* second half. For AR and (S)ARIMA, the last values (as dictated by the lags) of the first half were used to initiate generation. For AWSOM we again used as many of the last wavelet coefficients from each DWT level of the first half as were necessary to start applying the model equations. We should note that generating more than, say, 10 steps ahead is very rare: most methods in the literature [25] generate one step ahead, then obtain the correct value of X_{t+1}, and only then try to generate X_{t+2}. Nevertheless, our goal is to capture long-term behavior and AWSOM achieves this efficiently, unlike ARIMA.

5.1 Interpreting the models

Visual inspection. A "forecast" is essentially a by-product of any *generative* time series model: application of any model to generate a number of "future" values reveals precisely the trends and patterns captured by that model. In other words, synthesizing points based on the model is the simplest way for any user to get a quick, yet fairly accurate idea of what the trends are or, more precisely, what the *model* thinks

they are. Thus, what we expect to see (especially in a long-range forecast) is the *important* patterns that can be identified from the real data.

However, an expert user can extract even more precise information from the models. We will now explain how the "AWSOM language" can be fully interpreted.

Variance test. As explained in Appendix A, if the signal is self-similar, then the plot of log-power versus scale is linear.

Definition 2 (Variance diagnostic) *The log-power vs. scale plot is the* wavelet variance diagnostic plot *(or just* variance diagnostic*). In particular, the correlation coefficient ρ_α quantifies the relation. If the plot is linear (in a range of scales), the slope $\hat{\alpha}$ is the* self-similarity exponent *($-1 < \alpha < 0$, closer to zero the more bursty the series).*

A large value of $|\rho_\alpha|$, at least across several scales, indicates that the series component in those scales may be modeled using a fractional noise process with parameter dictated by α (see `Automobile`). However, we should otherwise be careful in drawing further conclusions about the behavior within these scales.

We should note that after the observation by [21], fractional noise processes and, in general, self-similar sequences have revolutionized network traffic modeling. Furthermore, self-similar sequences appear in atomic clock fluctuations, river minima, compressed video bit-rates [3, 23], to mention a few examples.

Wavelet variance (energy and power). The magnitude of variance within each scale serves as an indicator about which frequency components are the dominant ones in the sequence. To precisely interpret the results, we also need to take into account the fundamental uncertainty in frequencies (see Figure 13). However, the wavelet variance plot quickly gives us the general picture of important trends. Furthermore, it guides us to focus on AWSOM coefficients around frequencies with large variance.

AWSOM **coefficients.** Regardless of the energy within a scale, the AWSOM coefficients provide further information about the presence of trends, which cannot be deduced from the variance plots. In particular: **(a) Large intra-scale coefficients:** These capture patterns at certain frequencies, regardless of their energy presence. Furthermore, if the coefficients are not the same for all regression models at the same level, this is an indication of "seasonalities" within that scale and captures a different type of information about lower frequencies. **(b) Large inter-scale coefficients:** These occur when there are repeated bursts (see also Appendix D). The number of scales with large inter-scale coefficients depends on burst duration (short bursts have large bandwidth).

To summarize, the steps are: *(1)* Examine the variance diagnostic to identify sub-bands that correspond to a self-similar component. These may be modeled using a fractional noise process for generation purposes; for forecasting purposes they are just that: noise. *(2)* Examine the wavelet energy spectrum to quickly identify important sub-bands. *(3)* Examine AWSOM coefficients, primarily within and around the sub-bands identified during the second step.

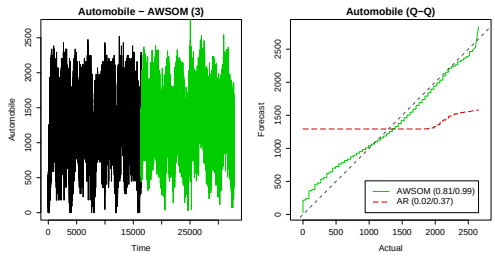


Figure 9: `Automobile`—*generation with fractional noise.*

5.2 Synthetic datasets

We present synthetic datasets to illustrate the basic properties of AWSOM, its behavior on several characteristic classes of sequences, and the principles behind interpreting the models. Applying the models to generate a number of "future" data points is the quickest way to see if each method captures long-term patterns.

`Triangle.` AR fails to capture anything, because the window is not large enough. SAR estimation (with no differencing, no MA component and only a manually pre-specified lag-256 seasonal component) fails completely. In fact, R segfaults after several minutes, even without using maximum-likelihood estimation (MLE). However, AWSOM captures the periodicity. The model visualization is omitted due to space constraints—see discussion on `Mix`.

`Mix.` AR is again confused and does not capture even the sinusoidal component. SAR estimation (without MLE) fails (R's optimizer returns an error, after several minutes of computation). Figure 12 shows the AWSOM coefficients. Due to space constraints, we show only the levels that correspond to significant variance. These illustrate the first point in the interpretation of AWSOM coefficients. We clearly see strong correlations in levels 6 and 8 (which correspond to the periods $2^6 = 64$ and $2^8 = 256$ of the series components). Note that the variance alone (see also Figure 13) is not enough to convey this information.

We also tried SAR$(0)\times(1)128$ on an impulse train of period 128. On 1024-point sequence, R takes over 4 minutes (on a signal with 64K points it did not complete in over one hour). However, AWSOM estimates the parameters (with 64K points) in approximately 50 seconds, although our prototype is implemented in Python.

5.3 Real datasets

Due to space constraints, we omit AWSOM coefficient visualizations. For the real datasets, we show the

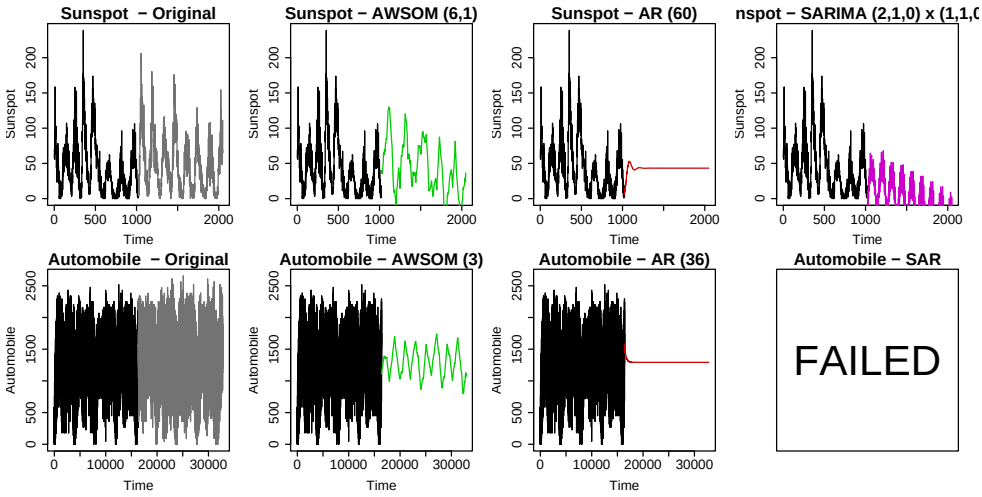


Figure 8: *Forecasts—real datasets. AR fails to detect any trend, while seasonal AR fails to complete or gives a wrong conclusion in* 260× *time.*

marginal distribution *quantile-quantile plots* (or *Q-Q plots*—see Figure 10 and Figure 9)[3].

`Sunspot.` This is a well-known dataset with a time-varying "period." AR again fails completely. SAR (without a MA component, much less MLE) takes 40 minutes to estimate. AWSOM (in Python) takes less than 9 seconds. SAR gives a completely fixed period and misses the marginal distribution (see Figure 10). AWSOM captures the general periodic trend, with a desirable slight confusion about the "period."

`Automobile.` This dataset has a strongly linear variance diagnostic in scales 1–6 (Figure 6). However, the lower frequencies contain the most energy (see Figure 11. This indicates we should focus at these scales. The lowest frequency corresponds to a daily periodicity (approximately 4000 points per day, or about 8 periods in the entire series) and next highest frequency corresponds to the morning and afternoon rush-hours.

In this series, high frequencies can be modeled by fractional noise. Figure 9 shows a generated sequence with fractional noise, as identified by AWSOM. The fractional difference parameter is estimated as $\hat{\delta} \equiv -\hat{\alpha}/2 \approx 0.276$ and the amplitude is chosen to match the total variance in those scales.

However, for unsupervised outlier detection, this is not necessary: what would really constitute an outlier would be, for instance, days that (a) do not follow the daily and rush-hour patterns, or (b) whose variance in the fractional noise scales is very different. This can be captured automatically by the series components in the appropriate frequency sub-bands that AWSOM identifies as a periodic component and bursty noise, respectively.

[3]These are the scatter plots of (x, y) such that $p\%$ of the values are below x in the real sequence and below y in the generated sequence. When the distributions are identical, the Q-Q plot coincides with the bisector of the first quadrant.

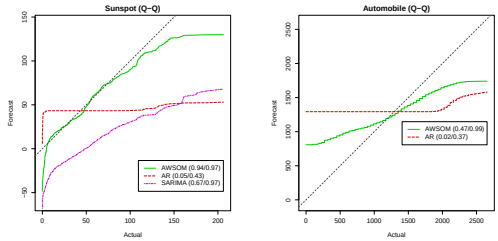


Figure 10: *Marginal Q-Q plots (slope and correlation coefficients in parentheses).*

6 Conclusions

Sensor networks are becoming increasingly popular, thanks to falling prices and increasing storage and processing power. We presented AWSOM, which achieves all of the following goals: *(1) Unsupervised operation:* once we decide the largest AWSOM order, no further intervention is needed: the sensor can be left alone to collect information. *(2) 'Any-time', one-pass* algorithm to incrementally update the patterns. *(3) Automatic* detection of arbitrary periodic components. *(4) Limited memory,* $O(\lg N)$. *(5) Simplicity:* AWSOM provides linear models. *(6) Power:* AWSOM provides information across several frequencies and can diagnose self-similarity and long-range dependence. *(7) Immediate outlier detection:* our method, despite its simplicity and its unsupervised operation, is able to do forecasting (directly, for the estimated model). We showed real and synthetic data, where our method captures the periodicities and burstiness, while manually selected AR (or even (S)ARIMA generalizations, which are not suitable for streams with limited resources) fails completely.

AWSOM is an important first step toward hands-off stream mining, combining simplicity with modeling power. Continuous queries are useful for evidence gathering and hypothesis testing *once* we know what we are looking for. AWSOM is the first method to

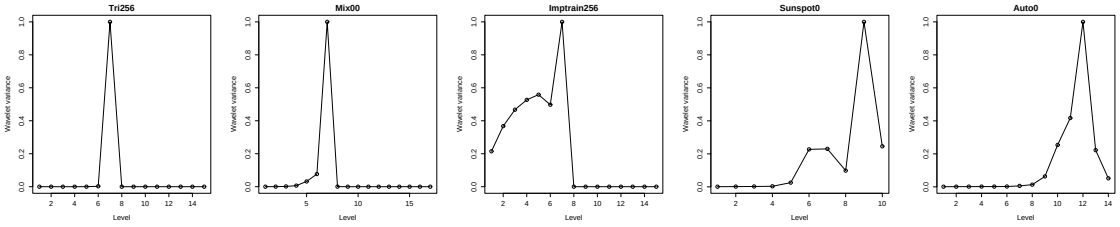


Figure 11: *Wavelet variances.*

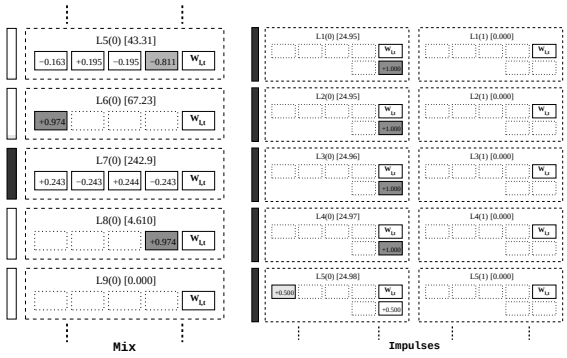


Figure 12: ***Mix*** *and* ***Impulses***—AWSOM *models. The bars on the left are proportional to the variance (see also Figure 11) and the numbers in brackets show the variance of the wavelet coefficients that correspond to each* AWSOM *equation. Blank boxes correspond to coefficients excluded by the model selection process.*

deal directly with the problem of unsupervised stream mining and pattern detection and fill the gap.

Acknowledgments. We thank Becky Buchheit for her help with the automobile traffic datasets.

References

[1] M. Akay, editor. *Time Frequency and Wavelets in Biomedical Signal Processing*. J. Wiley, 1997.

[2] A. Arasu, B. Babcock, S. Babu, J. McAlister, and J. Widom. Characterizing memory requirements for queries over continuous data streams. In *PODS*, 2002.

[3] J. Beran. *Statistics for Long-Memory Processes*. Chapman & Hall, 1994.

[4] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *J. Econometrics*, 31:307–327, 1986.

[5] P. Bonnet, J. E. Gehrke, and P. Seshadri. Towards sensor database systems. In *Proc. MDM*, 2001.

[6] P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer-Verlag, 2nd edition, 1991.

[7] A. Bulut and A. K. Singh. SWAT: Hierarchical stream summarization in large networks. In *Proc. 19th ICDE*, 2003.

[8] L. R. Carley, G. R. Ganger, and D. Nagle. Mems-based integrated-circuit mass-storage systems. *CACM*, 43(11):72–80, 2000.

[9] D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. B. Zdonik. Monitoring streams – a new class of data management applications. In *Proc. VLDB*, 2002.

[10] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-dimensional regression analysis of time-series data streams. In *Proc. VLDB*, 2002.

[11] M. Datar, A. Gionis, P. Indyk, , and R. Motwani. Maintaining stream statistics over sliding windows. In *Proc. SODA*, 2002.

[12] A. Dobra, M. N. Garofalakis, J. Gehrke, and R. Rastogi. Processing complex aggregate queries over data streams. In *Proc. SIGMOD*, 2002.

[13] C. Faloutsos. *Searching Multimedia Databases by Content*. Kluwer Academic Inc., 1996.

[14] M. N. Garofalakis and P. B. Gibbons. Wavelet synopses with error guarantees. In *Proc. SIGMOD*, 2002.

[15] J. Gehrke, F. Korn, and D. Srivastava. On computing correlated aggregates over continual data streams. In *Proc. SIGMOD*, 2001.

[16] R. Gencay, F. Selcuk, and B. Whitcher. *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*. Academic Press, 2001.

[17] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *Proc. VLDB*, 2001.

[18] S. Guha and N. Koudas. Approximating a data stream for querying and estimation: Algorithms and performance evaluation. In *Proc. ICDE*, 2002.

[19] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proc. ASPLOS-IX*, 2000.

[20] P. Indyk, N. Koudas, and S. Muthukrishnan. Identifying representative trends in massive time series data sets using sketches. In *Proc. VLDB*, 2000.

[21] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic. *IEEE Trans. on Networking*, 2(1):1–15, 1994.

[22] S. R. Madden, M. A. Shah, J. M. Hellerstein, and V. Raman. Continuously adaptive continuous queries over streams. In *SIGMOD Conf.*, 2002.

[23] D. B. Percival and A. T. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge Univ. Press, 2000.

[24] E. Riedel, C. Faloutsos, G. R. Ganger, and D. Nagle. Data mining on an OLTP system (nearly) for free. In *SIGMOD Conf.*, 2000.

[25] A. S. Weigend and N. A. Gerschenfeld. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison Wesley, 1994.

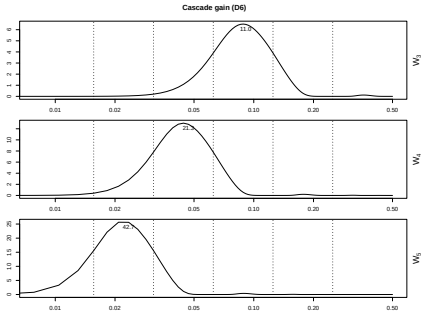


Figure 13: *Daubechies-6 cascade gain (levels 3–5).*

[26] B.-K. Yi, N. Sidiropoulos, T. Johnson, H. Jagadish, C. Faloutsos, and A. Biliris. Online data mining for co-evolving time sequences. *Proc. ICDE*, 2000.

[27] P. Young. *Recursive Estimation and Time-Series Analysis: An Introduction*. Springer-Verlag, 1984.

[28] D. Zhang, D. Gunopulos, V. J. Tsotras, and B. Seeger. Temporal aggregation over data streams using multiple granularities. In *Proc. EDBT*, 2002.

[29] R. Zuidwijk and P. de Zeeuw. Fast algorithm for directional time-scale analysis using wavelets. In *Proc. SPIE, Wavelet Applications in Signal and Image Processing VI*, volume 3458, 1998.

A More wavelet properties

Frequency properties. Wavelet filters employed in practice can only approximate an ideal bandpass filter, since they are of *finite* length L. The practical implications are that wavelet coefficients at level l correspond roughly to frequencies $[1/2^{l+1}, 1/2^l]$ (or, equivalently, periods $[2^l, 2^{l+1}]$ (see Figure 13 for the actual correspondence). This has to be taken into account for *precise* interpretation of AWSOM models by an expert.

Wavelet variance and self-similarity. The wavelet variance decomposes the variance of a sequence across scales. Due to space limitations, we mention basic definitions and facts; details can be found in [23].

Definition 3 (Wavelet variance) *If $\{W_{l,t}\}$ is the DWT of a series $\{X_t\}$ then the wavelet variance $\mathcal{V}_l$ is defined as $\mathcal{V}_l = \mathrm{Var}[W_{l,t}]$*

Under certain general conditions, $\hat{\mathcal{V}}_l = \frac{2^l}{N}\sum_{t=1}^{N/2^l} W_{l,t}$ is an *unbiased* estimator of $\mathcal{V}_l$. Note that the sum is precisely the energy of $\{X_t\}$ at scale l.

Definition 4 (Self-similar sequence) *A sequence $\{X_t\}$ is said to be self-similar following a pure power-law process if $\mathcal{S}_X(f) \propto |f|^\alpha$, where $-1 < \alpha < 0$ and $\mathcal{S}_X(f)$ is the SDF*[4]

It can be shown that $\mathcal{V}_l \approx 2\int_{1/2^{l+1}}^{1/2^l} \mathcal{S}_X(f)df$, thus if $\{X_t\}$ is self-similar, then $\boxed{\log \mathcal{V}_l \propto l}$ i.e., the plot of $\log \mathcal{V}_l$ versus the level l should be linear. In fact, slope of the log-power versus scale plot should be approximately equal to the exponent α. This fact and how to estimate $\mathcal{V}_l$ are what the reader needs to keep in mind.

[4]The *spectral density function (SDF)* is the Fourier transform of the auto-covariance sequence (ACVS) $S_{X,k} \equiv \mathrm{Cov}[X_t, X_{t-k}]$. Intuitively, it decomposes the variance into frequencies.

B Recursive Least Squares (RLS)

The least squares solution to an overdetermined system of equations $X\mathbf{b} = \mathbf{y}$ where $X \in \mathbb{R}^{m\times k}$ (measurements), $\mathbf{y} \in \mathbb{R}^m$ (output variables) and $\mathbf{b} \in \mathbb{R}^k$ (regression coefficients to be estimated) is given by the solution of $X^TX\mathbf{b} = X^T\mathbf{y}$. Thus, all we need for the solution are the projections

$$P \equiv X^TX \qquad \text{and} \qquad \mathbf{q} \equiv X^T\mathbf{y}$$

We need only space $O(k^2 + k) = O(k^2)$ to keep the model up to date. When a new row $\mathbf{x}_{m+1} \in \mathbb{R}^k$ and output y_{m+1} arrive, we can update $P \leftarrow P + \mathbf{x}_{m+1}\mathbf{x}_{m+1}^T$ and $\mathbf{q} \leftarrow \mathbf{q} + y_{m+1}\mathbf{x}_{m+1}$ In fact, it is possible to update the regression coefficient vector $\mathbf{b}$ without explicitly inverting P to solve $P\mathbf{b} = \mathbf{q}$ (see [27]).

C Model selection

We show how feature selection can be done from the data gathered online (i.e., P and $\mathbf{q}$ for each AWSOM equation).

C.1 Model testing

Lemma 4 (Square sum of residuals) *If $\mathbf{b}$ is the least-squares solution to the overdetermined equation $X\mathbf{b} = \mathbf{y}$, then*

$$s_n \equiv \sum_{i=1}^{n}(\mathbf{x_i}^T\mathbf{b} - y_i)^2 = \mathbf{b}^TP\mathbf{b} - 2\mathbf{b}^T\mathbf{q} + \mathbf{y}^2$$

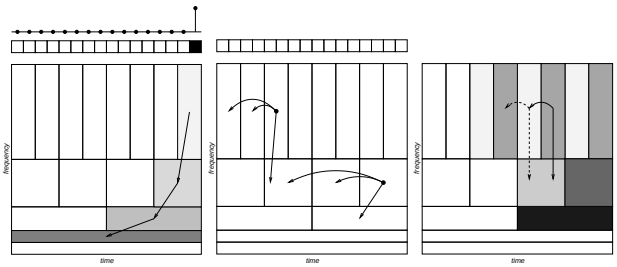
Proof Straightforward from the definition of s_n, which in matrix form is $s_n = (X\mathbf{b} - \mathbf{y})^2$. ■

Thus, besides P and $\mathbf{q}$, we only need to update $\mathbf{y}^2$ (a single number), by adding y_i^2 to it as each new value arrives. Now, if we select a subset $\mathcal{I} = \{i_1, i_2, \ldots, i_p\} \subseteq \{1, 2, \ldots, k\}$ of the k variables $x_1, x_2, \ldots, x_k$, then the solution $\mathbf{b}_\mathcal{I}$ for this subset is given by $P_\mathcal{I}\mathbf{b}_\mathcal{I} = \mathbf{q}_\mathcal{I}$ and the SSR by $s_n = \mathbf{b}_\mathcal{I}^TP_\mathcal{I}\mathbf{b}_\mathcal{I} - 2\mathbf{b}_\mathcal{I}\mathbf{q}_\mathcal{I} + \mathbf{y}^2$ where the subscript $\mathcal{I}$ denotes straight row/column selection (e.g., $P_\mathcal{I} = [p_{i_j,i_k}]_{i_j,i_k\in\mathcal{I}}$)

The *F-test (Fisher test)* is a standard method for determining whether a reduction in variance is statistically significant. The F-test is based on the sample variances, which can be computed *directly* from the SSR (Lemma 4). Although the F-test holds precisely (i.e., non-asymptotically) under normality assumptions, in practice it works well in several circumstances, especially when the population size is large. This is clearly the case with semi-infinite streams.

C.2 Model combination

If we split measurements x_i into two subsets X_1 and X_2 with corresponding outputs $\mathbf{y}_1$ and $\mathbf{y}_2$, then



(a) (b) (c)

Figure 14: *(a) Inter-scale correlations, intuition. (b,c) Illustration of* AWSOM$(1,1)$ *with* $L_i = 2$ *inactive levels. The shade of each wavelet coefficient corresponds to the model equation used to "predict" it. The unshaded wavelet coefficients correspond to initial conditions (i.e., with incomplete* AWSOM *window* $\mathcal{D}$*).*

the LS solution for both subsets combined is given by $b = (X^T X)^{-1} X^T y$ where $X = \left[X_1^T X_2^T\right]^T$ and $\mathbf{y} = [\mathbf{y}_1^T \mathbf{y}_2^T]^T$, i.e., $b = (X_1^T X_1 + X_2^T X_2)^{-1}(X_1^T \mathbf{y}_1 + X_2^T \mathbf{y}_2) = (P_1 + P_2)^{-1}(q_1 + q_2)$. Therefore, it is possible to combine sub-models when reducing the number of levels (effectively reducing $T \equiv 2^\lambda$). Model selection as presented above can be extended to include this case.

D Full model / Inter-scale correlations

Formally, our proposed method tries to fit models of the following form:

$$W_{l,t} = \sum_{(\delta l, \delta t) \in \mathcal{D}} \beta_{\delta l, \delta t} W_{l+\delta l, t/2^{\delta l} - \delta t} + \epsilon_{l,t} \qquad (2)$$

where $\mathcal{D}$ is a set of index offsets and $\epsilon_{l,t}$ is the usual error term. For example, in Figure 14(b), $\mathcal{D} = \{(0,1), (0,2), (1,0)\}$ and $W_{l,t} = \beta_{0,1} W_{l,t-1} + \beta_{0,2} W_{l,t-2} + \beta_{1,0} W_{l+1,t/2}$. The $\beta_{\delta l, \delta t}$ are called the AWSOM *coefficients.*

Definition 5 (AWSOM **order)** *The set of offsets is always of the form*

$$\begin{aligned} \mathcal{D} = \{ & (0,1),\ (0,2),\ \ldots,\ (0,n_0), \\ & (1,0),\ (1,1),\ (1,2),\ \ldots,\ (1,n_1-1), \\ & \ldots, \\ & (\lambda,0),\ \ldots,\ (\lambda,n_\lambda - 1)\ \} \end{aligned}$$

i.e., each wavelet coefficient is expressed as a function of the previous n_0 *wavelet coefficients on the same level,* n_1 *coefficients from one level below and so on. For a particular choice of* $\mathcal{D}$*, we use*

$$\text{AWSOM}(n_0, n_1, \ldots, n_\lambda)$$

to denote this instance of our model. We call $(n_0, \ldots, n_\lambda)$ *the model's* order*. The* total order *is the number of* AWSOM *coefficients* k *per equation, i.e.,* $k = \sum_{\delta l=0}^{\lambda} n_{\delta l}$ *and* λ *is called the* depth *of the model.*

For example, Figure 14(b) shows an AWSOM$(2,1)$ model. A fixed choice of $\mathcal{D}$ is sufficient for all signals. In most of our experiments we have used AWSOM$(6,4,2)$ (total order $k = 12$).

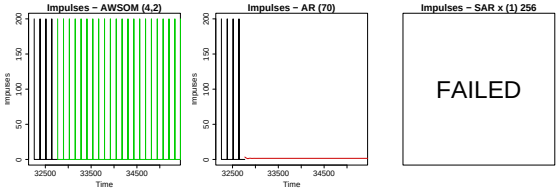


Figure 15: *Forecasts—*`Impulses`*. The signal consists of an impulse train (every 256 points), for a total of 64K points.*

Furthermore, we fit one equation per level (see Figure 14(b)), as long as the level contains enough wavelet coefficients to get a good fit. Thus, we fit one equation for every level $l < L_a$. These are the *active levels*, where L_a is the level that has no more than, say, $N_a = 16$ wavelet coefficients. For levels $l \geq L_a$ (the *inactive levels*), we can either keep the exact wavelet coefficients (which would be no more than $16 + 8 + \cdots + 1 = 31$ in the above case) and/or fit one more equation.

In other words, the number of inactive levels L_i is always fixed to, say, $L_i = 4$ and the number of active levels L_a grows as the stream size increases.

When fitting an AWSOM model with depth $\lambda \geq 1$, we also fit different equations depending on time location t. For instance, if we are using AWSOM$^1(n_0, 2)$, we should fit one equation for pairs $W_{l,2t}$ and $W_{l-1,t}$ and another for pairs $W_{l,2t+1}$ and $W_{l-1,t}$ (see Figure 14(c)). In general, we need 2^λ separate models to ensure that the inter-scale correlations λ levels down are not "shoehorned" into the same regression model.

To summarize, the full AWSOM model fits a number of equations:

$$W_{l,t} = \sum_{(\delta l, \delta t) \in \mathcal{D}} \beta_{\delta l, \delta t}^{l', t'} W_{l+\delta l, t-\delta t} \epsilon_{l,t} \qquad (3)$$

for $l' \leq L_a$ and $t' \equiv t \bmod T$, $0 \leq t' < T$. For example, if $T = 2$, we estimate one linear equation for each set of wavelet coefficients $W_{0,2i}$, $W_{0,2i+1}$, $W_{l,2i}$ and $W_{l,2k+1}$ ($l \geq 1$, $i \geq 0$). The significant advantage of this approach is that we can still easily update the AWSOM equations online, as new data values arrive. This is possible because the equation is selected based only on l and t for the new wavelet coefficient.

`Impulses`. This synthetic dataset (see Figure 15) illustrates how inter-scale correlations may help. AR fails to capture anything (again, too small window) and SAR estimation fails, while AWSOM captures the overall behavior. Figure 12 illustrates the second point in the interpretation of AWSOM coefficients. We clearly see repeated presence of bursts, with strong inter-scale correlations across *all* levels up to the impulse "period" (since the bursts have width one). Due to space constraints we show only those levels that correspond to the bursts. At level 5, information from the impulse "period" begins to enter in the wavelet coefficients (see also Figure 13). After level 7, the inter-scale correlations diminish in significance and the interpretation is similar to that for `Mix`.