

A Note on Algorithms for Arbitrarily-Spaced Time Series

Andreas Eckner

First version: April 2007

Current version: January 16, 2010

Abstract

This paper describes algorithms for efficiently calculating some of the time series operators presented in Eckner (2010). In particular, we show how to calculate moving averages (MAs) and exponentially moving averages (EWMA) in $O(n)$ time for arbitrarily-spaced time series. A web appendix provides an implementation of these algorithms in the programming language C.¹

Keywords: time series analysis, arbitrarily spaced series, irregularly spaced time series, moving average, exponentially weighted moving average, EWMA

1 Introduction

Müller (1991) and Dacorogna, Gençay, Müller, Olsen, and Pictet (2001) recommend, due to the computational simplicity, to use exponentially weighted moving averages (EWMA) as the main building block of time series operators for analyzing irregularly spaced time series data. In particular, Müller (1991) argues that the sequential computation of EWMA “is more efficient than the computation of any differently weighted MA.” This paper shows that many other time series operators can be calculated just as efficiently.

We use the notation $((t_n, X_n) : 1 \leq n \leq N(X))$ or $(X_{t_n} : 1 \leq n \leq N(X))$ to denote a time series X with observation times $\{t_1, \dots, t_{N(X)}\}$ and observations $(X_1, \dots, X_{N(X)})$, where $N(X)$ denotes the length of the time series. The algorithms in this paper have as input values (i) an array `values` containing the observation values, (ii) an array `times` containing the observation times, and (iii) a parameter τ describing the scale of a time series operator, such as the length of a moving average windows. The output is an array `out` of same length as the input arrays. Indices of arrays start at 1. Moreover, for brevity of the presentation we ignore memory allocation issues, numerical noise, as well as special cases for time series of length zero or one.

2 Moving Averages

Definition 1. For a time series X , we define three versions of the moving average (MA) of length $\tau > 0$. For $t \in \{t_1, \dots, t_{N(X)}\}$,

¹See <http://www.eckner.com/research.html>.

1. $MA^I(X, \tau)_t = \text{Avg}\{X_s : s \in [t, t - \tau)\}$,
2. $MA^{II}(X, \tau)_t = \frac{1}{\tau} \int_0^\tau X[t - s] ds$,
3. $MA^{III}(X, \tau)_t = \frac{1}{\tau} \int_0^\tau X[t - s]_{\text{lin}} ds$,

where in all three cases the observation times of the input and output time series are identical. Here $X[s]$ denotes the most recent observation value of X at or before time s , while $X[s]_{\text{lin}}$ denotes the linearly interpolated value of X at time s .

See Eckner (2010), Section 7 for a detailed description and motivation of these three moving average versions. Intuitively, using MA^I is most appropriate for analyzing truly discrete *events*, for example, for calculating the average number of casualties per deadly car accident over the past twelve months. Or the average number of IBM shares traded per executed market order during the past 30 minutes. MA^{II} is most appropriate for analyzing truly discrete observation *values*, for example, for calculating the average FED funds target rate² over the past three years. In this case, it is desirable to weigh historical target rates by the amount of time such target levels remained unchanged. Finally, MA^{III} is most appropriate for estimating the rolling average value of a continuous time stochastic processes with observation times that are independent of the observation values.

2.1 MA^I

The moving average MA^I can be calculated efficiently by keeping track of (i) the number and (ii) sum of observation values in a window of width τ that moves forward in time. Whenever a new observation enters or leaves the time window, these two values are updated.³

The pseudocode of the algorithm is as follows:

```

left = 1, right = 0, rollsum = 0;
while (right < N(X) {
    right = right + 1, rollsum = rollsum + X[right];
    while ((left < right) and (T[left+1] < T[right] - tau))
        rollsum = rollsum - X[left], left = left + 1;
    out[right] = rollsum / (right - left + 1);
}

```

Remark. The algorithm above can be easily modified to calculate (i) the number, and (ii) sum of observations in a rolling time window. See the web appendix for an implementation in the programming language \mathcal{C} .

2.2 MA^{II}

An efficient calculation of the moving average MA^{II} can be achieved in a similar manner to MA^I . However, now the spacing of observations leaving and entering the rolling time window needs to be taken into account. In order to avoid various special cases in the algorithm, an auxiliary

²The FED funds target rate is the desired interest rate (by the FED) at which depository institutions (such as banks) lend balances at the Federal Reserve to other depository institutions overnight. See <http://www.federalreserve.gov/fomc/fundsrate.htm> for details.

³To avoid numerical noise due to a potentially large number of additions and subtractions, it is recommended to occasionally calculate the sum of time series observation values in the rolling window from scratch.

observation is added to the time series X under investigation. Specifically, for calculating $\text{MA}^{\text{II}}(X, \tau)$ with $\tau > 0$, the algorithm first inserts a new observation at time $t = \min T(X) - \tau$ with value equal to X_1 :

```

j = N(X);
while (j > 1)
    values[j+1] = values[j], times[j+1] = times[j], j = j - 1;
times[1] = times[2] - tau;

```

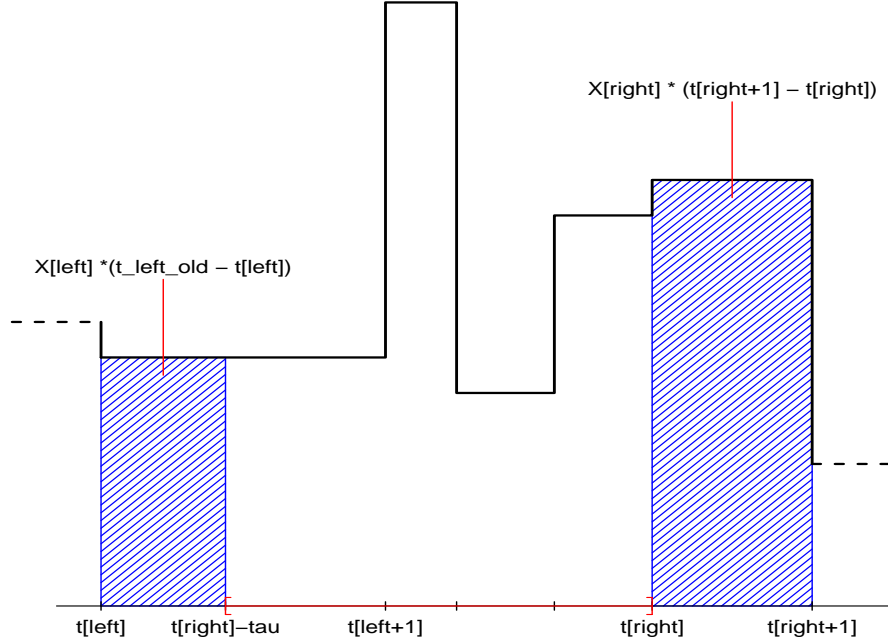


Figure 1: Integration areas involved in the calculation of the moving average $\text{MA}^{\text{II}}(X, \tau)$. The shaded area on the left-hand side has to be added to the variable *rollsum* in order to get an integration area with endpoints coinciding with observation times. The shaded area on the right-hand side has to be added to the variable *rollsum* in order to expand the rolling time window to the right.

Figure 1 highlights important steps in the main part of the algorithm, which is as follows:

```

left = 1; right = 2; rollsum = values[2] * tau; out[1] = values[2];
while (right < N(X)) {
    // Expand interval on right end
    rollsum = rollsum + values[right] * (times[right+1] - times[right]);

    // Add truncated left end of interval back in
    t_left_old = times[right] - tau;
    rollsum = rollsum + values[left] * (t_left_old - times[left]);
}

```

```

// Shrink interval on left end
while (times[left+1] <= t_left_new) {
    rollsum = rollsum - values[left] * (times[left+1] - times[left]);
    left = left + 1;
}

// Truncate left end of interval
t_left_new = times[right+1] - tau;
rollsum = rollsum - values[left] * (t_left_new - times[left]);

// Calculate MA value for current time window
out[right] = rollsum / tau;
right = right + 1;
}

```

This algorithm implicitly assumes that before the first observation time t_1 , the time series X is equal to the first observation value X_1 . The algorithm therefore might give more weight to the first observation than other observations during an initial ramp-up period of length τ . Nevertheless, whatever ramp-up assumption is used for calculating the moving average during the time interval $[t_1, t_1 + \tau]$, the value of $\text{MA}^{\text{II}}(X, \tau)$ after time $t_1 + \tau$ does not depend on this assumption.

2.3 MA^{III}

For calculating the moving average MA^{III} , the algorithm again first inserts a new observation at time $t = \min T(X) - \tau$ with value equal to X_1 :

```

j = N(X);
while (j > 1)
    values[j+1] = values[j], times[j+1] = times[j], j = j - 1;
times[1] = times[2] - tau;

```

Figure 2 demonstrates the calculation of some of the intermediate quantities, such as `area`, in the main part of the algorithm, which is as follows:

```

left = 1; right = 2; rollsum = values[2] * tau; out[1] = values[2];
while (right < N(X)) {
    // Expand interval on right end
    rollsum = rollsum + (values[right+1] + values[right])/2 *
        (times[right+1] - times[right]);

    // Add truncated left end of interval back in
    t_left_old = times[right] - tau;
    length = t_left_old - t[left];
    height = values[left] + (values[left+1] - values[left]) *
        length / (times[left+1] - times[left]);
    area = length * (values[left] + height)/2;
}

```

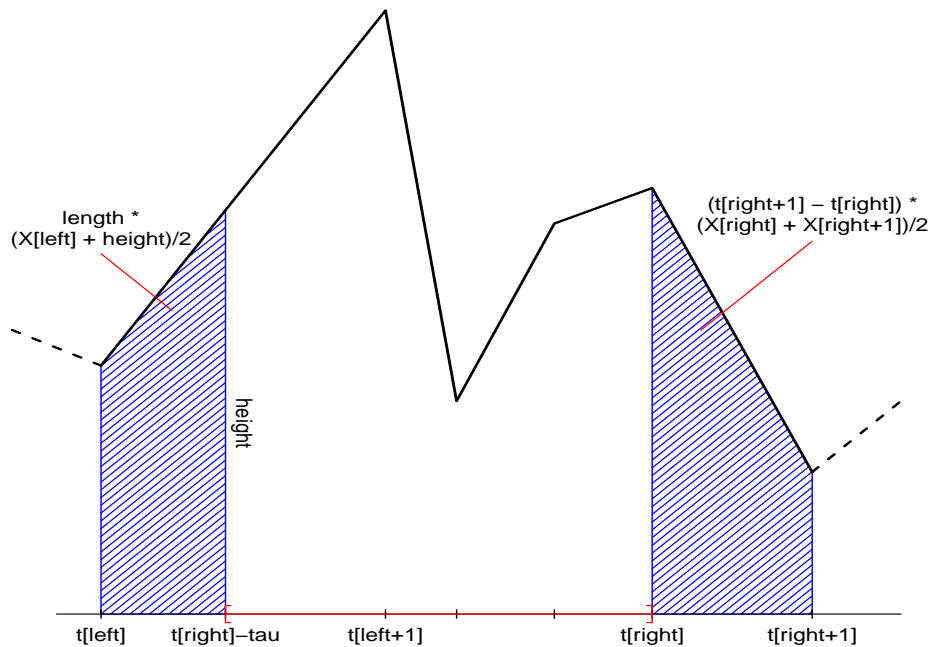


Figure 2: Integration areas involved in the calculation of the moving average $\text{MA}^{\text{III}}(X, \tau)$. The shaded area on the left-hand side has to be added to the variable *rollsum* in order to get an integration area with endpoints coinciding with observation times. The shaded area on the right-hand side has to be added to the variable *rollsum* in order to expand the rolling time window to the right.

```

rollsum = rollsum + area;

// Shrink interval on the left
while (times[left+1] <= t_left_new) {
    rollsum = rollsum - (values[left] + values[left+1]) / 2 *
        (times[left+1] - times[left]);
    left = left + 1;
}

// Truncate left end of interval
t_left_new = times[right+1] - tau;
length = t_left_new - t[left];
height = values[left] + (values[left+1] - values[left]) *
    length / (times[left+1] - times[left]);
area = length * (values[left] + height) / 2;
rollsum = rollsum - area;

// Calculate MA value for current time window

```

```

    out[right] = rollsum / tau;
    right = right + 1;
}

```

As before, the algorithm implicitly assumes that before the first observation time t_1 , the time series X is equal to the first observation value X_1 . The effect of this or any similar assumption disappears after an initial ramp-up period of length τ .

3 Exponentially Weighted Moving Averages

Definition 2. For a time series X , we define three versions of the exponentially weighted moving average (EWMA) of length $\tau > 0$. For $t \in \{t_1, \dots, t_{N(X)}\}$,

1. $\text{EWMA}^{\text{I}}(X, \tau)_t = (1 - e^{-\Delta t_n / \tau}) X_{t_n} + e^{-\Delta t_n / \tau} \text{EWMA}^{\text{I}}(X, \tau)_{t_{n-1}}$,
2. $\text{EWMA}^{\text{II}}(X, \tau)_t = \frac{1}{\tau} \int_0^\infty X[t-s] e^{-s/\tau} ds$,
3. $\text{EWMA}^{\text{III}}(X, \tau)_t = \frac{1}{\tau} \int_0^\infty X[t-s]_{\text{lin}} e^{-s/\tau} ds$,

where in all three cases the observation times of the input and output time series are identical.

See [Eckner \(2010\)](#), Section 7 for a detailed description and motivation of these three EWMA versions. Note that EWMA^{I} is the most widely used version by practitioners and commonly referred to as *the* exponentially weighted moving average.

3.1 EWMA^{I}

The EWMA^{I} can be calculated in a simple recursive manner:

```

j = 1; out[1] = values[1];
while (j < N(X)) {
    j = j + 1;
    w = exp(-tau * (times[j] - times[j-1]));
    out[j] = out[j-1] * w + values[j] * (1-w);
}

```

3.2 EWMA^{II}

It is easy to show that Definition 2 of the EWMA^{II} is equivalent to

$$\text{EWMA}^{\text{II}}(X, \tau)_{t_n} = \left(1 - e^{-\Delta t_n / \tau}\right) X_{t_{n-1}} + e^{-\Delta t_n / \tau} \text{EWMA}^{\text{II}}(X, \tau)_{t_{n-1}},$$

for $1 < n \leq N(X)$.⁴ Hence, the $\text{EWMA}^{\text{II}}(X, \tau)$ can be calculated in a simple recursive manner as well:

⁴In particular, the EWMA^{II} of a time series with infinite history is identical to the EWMA^{I} of the back-shifted time series. That is, $\text{EWMA}^{\text{II}}(X, \tau) = \text{EWMA}^{\text{I}}(B(X), \tau)$, where $B(X) = ((t_n, X_{n-1}) : n \in \mathbb{Z})$.

```

j = 1; out[1] = values[1];
while (j < N(X)) {
  j = j + 1;
  w = exp(-tau * (times[j] - times[j-1]));
  out[j] = out[j-1] * w + values[j-1] * (1-w);
}

```

3.3 EWMA^{III}

The EWMA^{III} can again be calculated in a recursive manner, see [Eckner \(2010\)](#) for a derivation of the recursion:

```

j = 1; out[1] = values[1];
while (j < N(X)) {
  j = j + 1;
  tmp = (times[j] - times[j-1]) / tau;
  w = exp(-tmp);
  w2 = (1 - w) / tmp;
  out[j] = out[j-1] * w + values[j] * (1 - w2) + values[j-1] * (w2 - w);
}

```

4 More General Operators

This section briefly discussed how to generalize some of the time series operators discussed above and how to combine operators to get new ones.

4.1 Convolution Operators

Recall the following definition of convolution operators (see [Gilles Zumbach \(2001\)](#), [Dacorogna, Gençay, Müller, Olsen, and Pictet \(2001\)](#), and [Eckner \(2010\)](#) for details):

Definition 3. *For a time series X and function f on $\mathbb{R} \times \mathbb{R}_+$ satisfying suitable measurability and integrability conditions, the last-observation convolution $X * f$ of X with f is given by*

$$(X * f)_t = \int_0^\infty f(X[t-s], s) ds, \quad t \in \{t_1, \dots, t_{N(X)}\}.$$

The linear convolution of $X *^{\text{lin}} f$ of X with f is given by

$$(X *^{\text{lin}} f)_t = \int_0^\infty f(X[t-s]_{\text{lin}}, s) ds, \quad t \in \{t_1, \dots, t_{N(X)}\},$$

where $X[s]_{\text{lin}}$ for $t_1 \leq s \leq t_{N(X)}$ is the linearly interpolated value of X at time s . In both cases, the observation times of the input and output time series are identical.

The algorithm for MA^{II} in Section 2 can be easily modified to allow calculating convolutions with densities of the form

$$f(x, t) = g(x) \mathbf{1}_{\{0 \leq t \leq \tau\}}, \quad (4.1)$$

where $\tau > 0$ and g is a real-valued function. For example, $g(x) = x^k/\tau$ for $k \in \mathbb{N}$ allows to calculate the rolling k th time series moment, denoted by $m(X, \tau, k)$. In particular, $k = 1$ amounts to calculating the MA^{II} .

The algorithm for MA^{III} can be adapted to kernels of the form (4.1), but may require time-consuming numerical integration of the triangular areas in Figure 1. Alternatively, the function g in (4.1) can first be applied to the series values of X , and then the moving average MA^{III} is calculated of the transformed time series.

4.2 Combining Operators

More complex operators can be constructed by taking (linear) combinations of simpler time series operators. As a consequence, these new operators can be calculated in $O(N(X))$ time as well. For example, the double EWMA difference

$$\text{dEWMA}^n(X, \tau_1, \tau_2) = \text{EWMA}^n(X, \tau_1) - \text{EWMA}^n(X, \tau_2),$$

for $n \in \{\text{I,II,III}\}$ is a commonly used technical indicator for financial time series.

The rolling variance of a time series can be calculated as

$$\sigma^2(X, \tau) = m(X, \tau, 2) - (\text{MA}^{\text{II}}(X, \tau))^2,$$

where X^k for a time series X is short-hand notation for taking the k th moment of the individual time series observations. The calculation of higher-order centered moments is similar.

Finally, convolutions $X * f$ with density of the form $f(x, t) = g(x)h(t)$ with measurable functions $g : \mathbb{R} \rightarrow \mathbb{R}$ and $h : \mathbb{R}_+ \rightarrow \mathbb{R}$ can be approximated with arbitrary precision by a linear combination of moving averages MA^{II} of the transformed time series $g(X)$.

5 Conclusion

We have shown that a wide range of time series operators for arbitrarily-spaced data can be calculated in linear time, that is, in $O(N(X))$. These operators in turn may serve as building blocks for more complicated operators.

References

- Dacorogna, M. M., R. Gençay, U. A. Müller, R. B. Olsen, and O. V. Pictet (2001). *An Introduction to High-Frequency Finance*. Academic Press.
- Eckner, A. (2010). Arbitrarily-spaced time series analysis. Work in progress.
- Gilles Zumbach, U. A. M. (2001). Operators on inhomogeneous time series. *International Journal of Theoretical and Applied Finance* 4, 147–178.
- Müller, U. A. (1991). Specially weighted moving averages with repeated application of the ema operator. Working Paper, Olsen and Associates, Zurich, Switzerland.