Exponential moving average filter

Marijn Tamis

May 22, 2014

Last time

We saw that we could make the fixed time step algorithm work with a variable time step:

Fixed time step algorithm formula

$$v_2 = v_1 c + v_t(1 - c)$$

Variable time step algorithm

$$v_2 = v_1 \left(e^{-\lambda \Delta t}\right) + v_t \left(1 - e^{-\lambda \Delta t}\right)$$

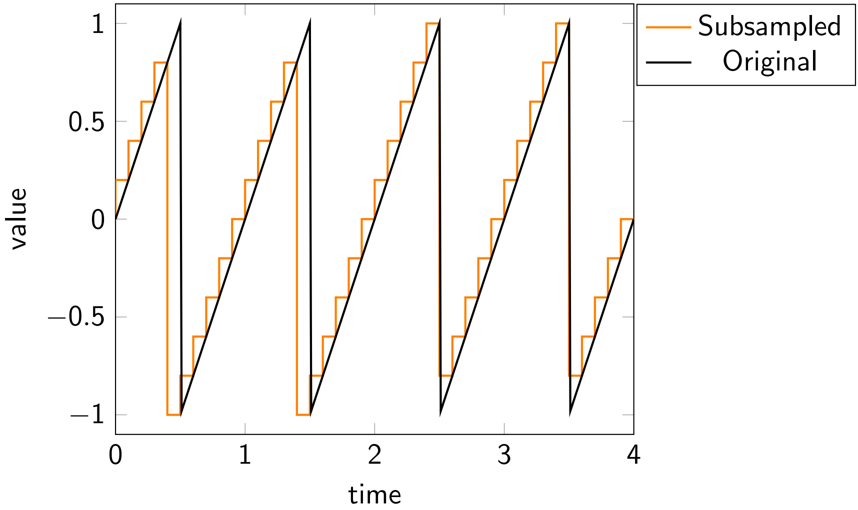
Exponential decay

$$e^{-\lambda \Delta t_f} = c$$

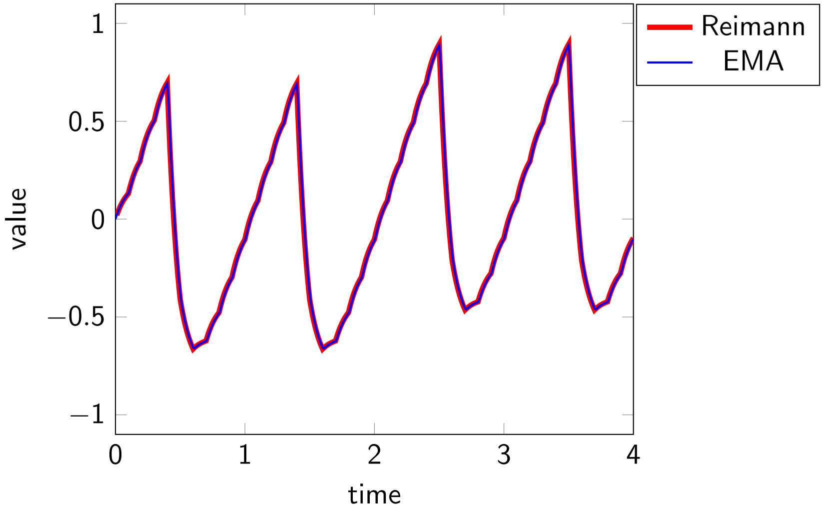
$$-\lambda = \frac{\ln(c)}{\Delta t_f}$$

Note: Δt_f is the delta time for the fixed frame rate algorithm

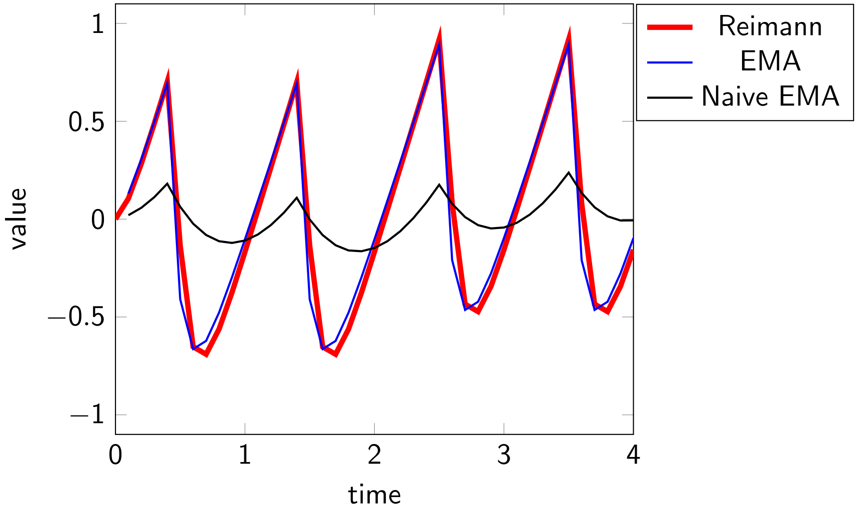
Input function



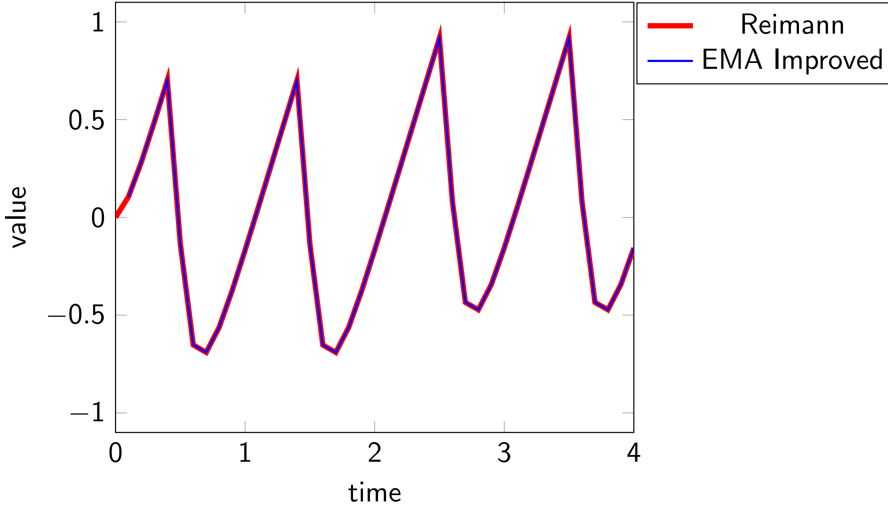
Filter with small Δt



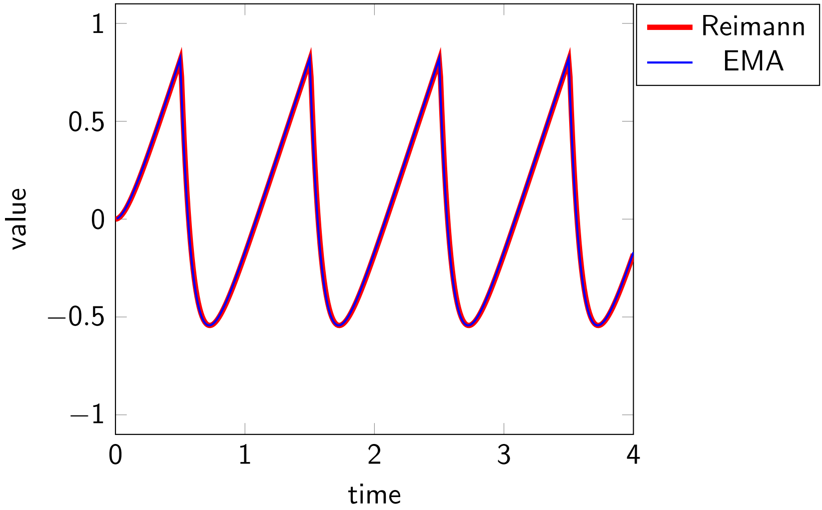
Filter with big Δt



Filter with big Δt



Filter on original data with small Δt



What is a filter

In signal processing, a filter is a device or process that removes from a signal some unwanted component or feature.

-From Wikipedia, the free encyclopedia

What is a filter

- We have a data set of n points $x_0, x_1, \ldots, x_{n-1}$.
- Simple moving average at time t is $\text{SMA}(t)$.
- $\text{SMA}(t)$ is the average of all points up to t.

$$\text{SMA}(t) = \frac{x_0 + x_1 + \ldots + x_t}{t} = \frac{1}{t}\sum_{i=0}^{t} x_i$$

- This is a low pass filter where each sample has an equal weight.
- If we want to only take the last k samples in to account we can do $\text{SMA}(t) - \text{SMA}(t-k)$.

Note: It gets more complicated if the points are not equally spaced, or if there are more points between x_t and x_{t+1}.

What is a filter

- If we want to weigh each sample differently we get an Weighted moving average.
- We have weights for every point $w_0, w_1, \ldots, w_{n-1}$.
- The Weighted Moving Average is defined as:

$$\text{WMA}(t) = \frac{x_0 w_0 + x_1 w_1 + \ldots + x_t w_t}{w_0 + w_1 + \ldots + w_t}$$

- Or more general:

$$\text{WMA}(t) = \frac{1}{w_{\text{total}}} \sum_{i=0}^{t} (x_i w_i)$$

where w_{total} is the sum of all weights.

What is a filter

- It is often useful to align the wights with t.
- You have already done this with Gaussian blur approximations.
- Blur kernel defines the weights.
- Blur kernel moves with t (position).
- Gaussian blur needs to sample ahead.

What is a filter

- It is often useful to align the wights with t.

$$\text{WMA}(t) = \frac{1}{w_{\text{total}}} \sum_{i=0}^{t} (x_i w_{(i-t)})$$

- For weights $w_{-t} \dots w_0$.

What is a filter

- Calculating SMA(t) for n data points takes

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

  steps.
- We evaluate SMA(t) for each data point.
- SMA(t) Needs all samples up to t.
- $O(n^2)$ is slow.

What is a filter

- Calculating SMA(t) for n data points takes

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

  steps.
- We evaluate SMA(t) for each data point.
- SMA(t) Needs all samples up to t.
- $O(n^2)$ is slow.
- But we can do better!

What is a filter

$$\text{SMA}(t+1) = \frac{\text{SMA}(t)t + x_{t+1}}{t+1}$$

$$\frac{1}{t+1}\sum_{i=0}^{t} x_i = \frac{(\frac{1}{t+1}\sum_{i=0}^{t} x_i)t + x_{t+1}}{t+1} = \frac{\sum_{i=0}^{t} x_i + x_{t+1}}{t+1}$$

- $O(n)$

- Or if we only want the last k samples:

$$\mathrm{SMA}(t+1) - \mathrm{SMA}(t+1-k) = \mathrm{SMA}(t) + \frac{x_{t+1}}{k} - \frac{x_{t+1-k}}{k}$$

$$\frac{1}{k}\sum_{i=t+1-k}^{t} x_i = \text{etc.}$$

What is a filter

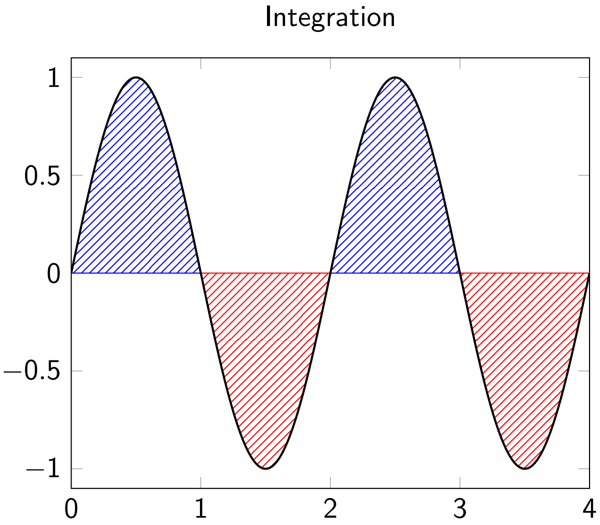
- Same optimization can be done for WMA(t).
- Always need to keep the last k samples for FIR filters.

What is a filter

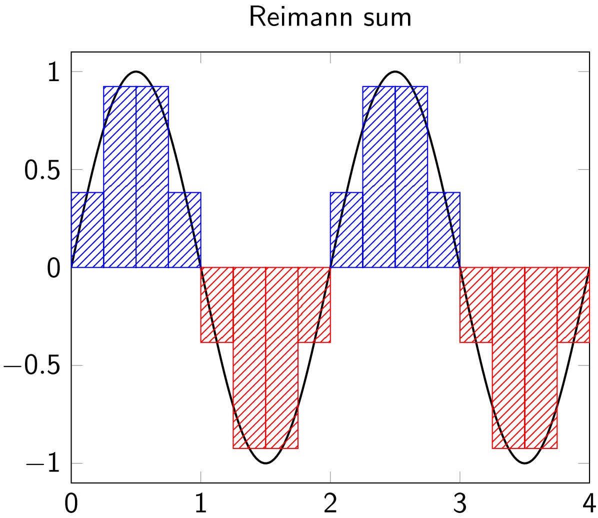
- How do we calculate the Simple Moving Average for a function?

- How do we calculate the Simple Moving Average for a function?
- We integrate!

What is a filter



What is a filter



What is a filter

- SMA(t) of f(x):

$$\mathrm{SMA}(t) = \frac{1}{t}\int_0^t \mathrm{f}(x)dx$$

- See the similarities with $\frac{1}{t}\sum_{i=0}^{t} x_i$.
- The sum has been replaced with an integral.

What is a filter

- We can also do this for the Weighted Moving Average if we have a weight function w(x).
- We define $w_{\text{total}}(t) = \int_0^t w(x)dx$
- WMA(t) of f(x) is:

$$\text{WMA}(t) = \frac{1}{w_{\text{total}}} \int_0^t \text{f}(x)\text{w}(x)dx$$

- See the similarities with $\frac{1}{w_{\text{total}}} \sum_{i=0}^{t} (x_i w_{(i)})$.

What is a filter

- We can also align w(x) with t:

$$\mathrm{WMA}(t) = \frac{1}{w_{\mathrm{total}}} \int_0^t \mathrm{f}(x)\mathrm{w}(x - t)dx$$

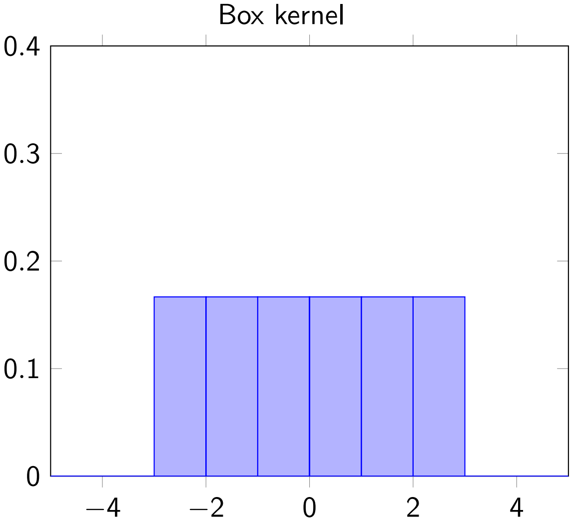
What is a filter

- This is very similar to convolution

$$(\mathrm{f}(x) * \mathrm{g}(x))(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} \mathrm{f}(\tau)\mathrm{g}(t-\tau)d\tau$$

- And it is the same if $\mathrm{g}(x) = 0$ for $x > 0$ and if $\mathrm{f}(x) = 0$ for $x < 0$

What does a filter look like

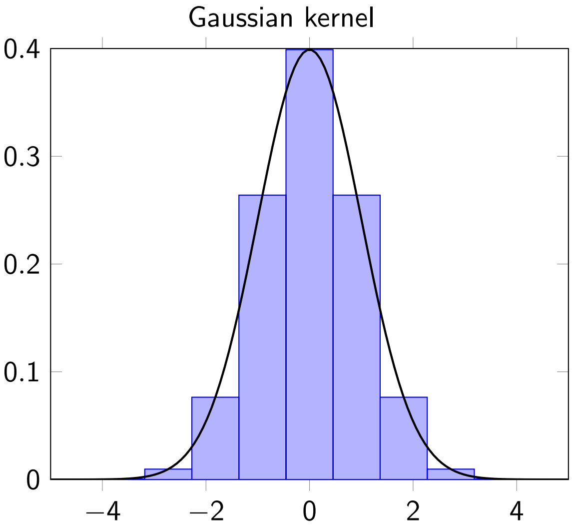


What does a filter look like

Box kernel

$$w(x) = \begin{cases} \frac{1}{s}, & \text{if } |x| \leq \frac{s}{2}, \\ 0, & \text{if } |x| > \frac{s}{2}, \end{cases}$$

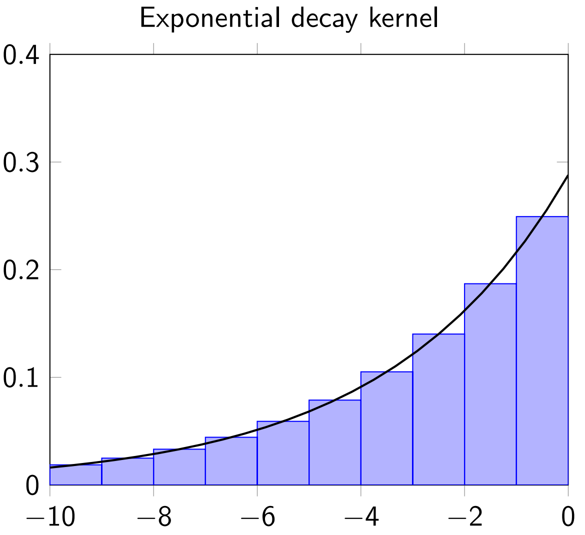
What does a filter look like



Gaussian kernel

$$\mathrm{w}(x) = e^{\frac{-x^2}{2c^2}} \left(\int_{-s}^{s} e^{\frac{-x^2}{2c^2}} dx \right)^{-1} \text{for } |x| \leq s$$

What does a filter look like



What does a filter look like

Exponential decay kernel

$$w(x) = e^{-\lambda x}(-\lambda) \text{ for } x < 0$$

As

$$W(x) = \int w(x)dx = e^{-\lambda x}$$

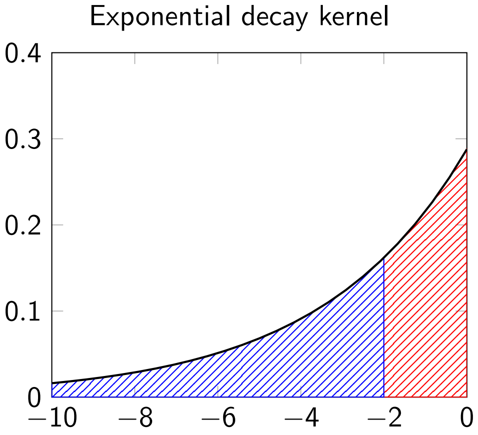
$$\int_{-\infty}^{0} w(x) = W(0) - W(-\infty) = 1 - e^{\lambda\infty} = 1$$

Calculating the serial weights

- Scale $\mathrm{w}(t)$ by $\frac{1}{w_{\text{total}}}$ to omit the division in the algorithm.

Calculating the serial weights

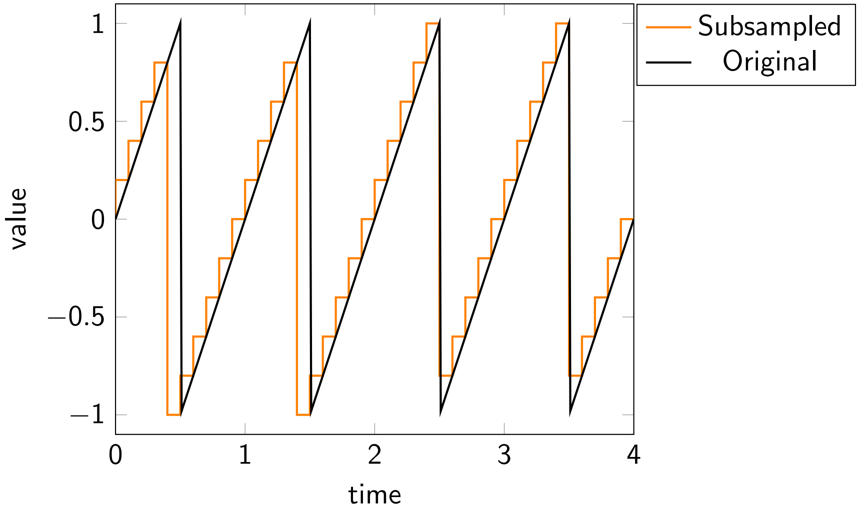
- Note that $\int_{-t}^{0} \mathrm{w}(x)dx = 1 - e^{\lambda t}$ (red).
- And that $\int_{-\infty}^{-t} \mathrm{w}(x)dx = e^{\lambda t}$ (blue).



Calculating the serial weights

- $\mathrm{EMA}(t)$ is the Exponential Moving Average at t.
- Kernel moves with t.
- So $(\mathrm{f}(x) * \mathrm{w}(x))(t) = \mathrm{EMA}(t)$ where $\mathrm{w}(x)$ is the exponential decay kernel.
- We know $\mathrm{EMA}(t) = 1$ for $\mathrm{f}(x) = 1$.
- So lets assume that $\mathrm{EMA}(t + \Delta t) = \mathrm{EMA}(t)e^{\lambda t} + \mathrm{f}(x)\left(1 - e^{\lambda t}\right)$ given that $\mathrm{f}(x) = y$ for $t < x \leq t + \Delta t$.

Input function



Why exponential decay

- Lets substitute some iterations manually.
- We say that $v_1 = 0$ for convenience.

1. $v_2 = v_{t1}(1 - c)$
2. $v_3 = v_2 c + v_{t2}(1 - c)$

Why exponential decay

- Lets substitute some iterations manually.
- We say that $v_1 = 0$ for convenience.

1. $v_2 = v_{t1}(1 - c)$
2. $v_3 = v_2 c + v_{t2}(1 - c) = v_{t1}(1 - c)c + v_{t2}(1 - c)$

Why exponential decay

- Lets substitute some iterations manually.
- We say that $v_1 = 0$ for convenience.

1. $v_2 = v_{t1}(1 - c)$
2. $v_3 = v_2 c + v_{t2}(1 - c) = v_{t1}(1 - c)c + v_{t2}(1 - c)$
3. $v_4 = v_3 c + v_{t3}(1 - c) = (v_{t1}(1 - c)c + v_{t2}(1 - c))c + v_{t3}(1 - c) = v_{t1}(1 - c)c^2 + v_{t2}(1 - c)c + v_{t3}(1 - c)$

Why exponential decay

- Lets substitute some iterations manually.
- We say that $v_1 = 0$ for convenience.

1. $v_2 = v_{t1}(1-c)$
2. $v_3 = v_2 c + v_{t2}(1-c) = v_{t1}(1-c)c + v_{t2}(1-c)$
3. $v_4 = v_3 c + v_{t3}(1-c) = (v_{t1}(1-c)c + v_{t2}(1-c))c + v_{t3}(1-c) = v_{t1}(1-c)c^2 + v_{t2}(1-c)c + v_{t3}(1-c)$
4. $v_5 = v_4 c + v_{t4}(1-c) = (v_{t1}(1-c)c^2 + v_{t2}(1-c)c + v_{t3}(1-c))c + v_{t4}(1-c) = v_{t1}(1-c)c^3 + v_{t2}(1-c)c^2 + v_{t3}(1-c)c + v_{t4}(1-c)$

Why exponential decay

- Set the results next to each other and we see a pattern.

1. $v_2 = v_{t1}(1-c)$
2. $v_3 = v_{t1}(1-c)c + v_{t2}(1-c)$
3. $v_4 = v_{t1}(1-c)c^2 + v_{t2}(1-c)c + v_{t3}(1-c)$
4. $v_5 = v_{t1}(1-c)c^3 + v_{t2}(1-c)c^2 + v_{t3}(1-c)c + v_{t4}(1-c)$

Sum notation of naive algorithm

$$v_x = (1-c)\sum_{i=1}^{x}\left(v_{ti}c^{(x-i)}\right)$$

- Note the increasing exponent.

Why exponential decay

- An exponential decay is usually defined with e as base.

Sum notation of naive algorithm

$$v_x = (1 - c) \sum_{i=1}^{x} \left(v_{ti} c^{(x-i)} \right)$$

- If $e^x = y$ then $\ln(e^x) = \ln(y) = x$
- So $e^{\ln(y)} = y$

Why exponential decay

- An exponential decay is usually defined with e as base.

Sum notation of naive algorithm

$$v_x = (1 - c) \sum_{i=1}^{x} \left(v_{ti} c^{(x-i)} \right)$$

- If $e^x = y$ then $\ln(e^x) = \ln(y) = x$
- So $e^{\ln(y)} = y$

Sum notation of naive algorithm with base e

$$v_x = (1 - c) \sum_{i=1}^{x} \left(v_{ti} e^{\ln(c^{(x-i)})} \right)$$

Why exponential decay

Sum notation of naive algorithm with base e

$$v_x = (1-c)\sum_{i=1}^{x}\left(v_{ti}e^{\ln(c^{(x-i)})}\right)$$

- We want to get x out of the $\ln(\ldots)$ function.
- Properties of log says that $\log(c^x) = x\log(c)$.

Sum notation of naive algorithm with base e

$$v_x = (1-c)\sum_{i=1}^{x}\left(v_{ti}e^{\ln(c)(x-i)}\right)$$

Why exponential decay

- We know that $e^{\ln cx} e^{\ln cx} = e^{\ln cx2}$.
- And more importantly $e^{\ln cx} e^{\ln cy} = e^{\ln c(x+y)}$.

Sum notation of naive algorithm with base e

$$v_x = (1 - c) \sum_{i=1}^{x} \left(v_{ti} e^{\ln(c)(x-i)} \right)$$

Serial sum notation of naive algorithm with base e

$$v_{x+1} = (1 - c) \sum_{i=1}^{x} \left(v_{ti} e^{\ln(c)(x-i+1)} \right) + v_{tx} e^{\ln(c)(0)}$$

Serial sum notation of naive algorithm with base e

$$v_{x+1} = (1 - c) \sum_{i=1}^{x} \left(v_{ti} e^{\ln(c)(x-i)} \right) e^{\ln(c)(1)} + (1 - c) v_{tx}$$

Why exponential decay

- Finally $O(n)$, (or $O(1)$ for the next sample).
- Without keeping k samples in memory.

Serial sum notation of naive algorithm with base e

$$v_{x+1} = (1-c)\sum_{i=1}^{x}\left(v_{ti}e^{\ln(c)(x-i)}\right)e^{\ln(c)(1)} + (1-c)v_{tx}$$

Serial sum notation of naive algorithm with base e

$$v_{x+1} = v_x e^{\ln(c)} + (1-c)v_{tx}$$

Serial sum notation of naive algorithm with base e

$$v_{x+1} = v_x e^{\ln(c)} + (1-e^{\ln(c)})v_{tx}$$

Why exponential decay

- We end up with the formula from last presentation.

Serial sum notation of naive algorithm with base e

$$v_{x+1} = v_x e^{\ln(c)} + (1 - e^{\ln(c)}) v_{tx}$$

Variable time step algorithm

$$v_2 = v_1 \left(e^{-\lambda \Delta t} \right) + v_t \left(1 - e^{-\lambda \Delta t} \right)$$

Interesting links

- http://en.wikipedia.org/wiki/Moving_average
- http://en.wikipedia.org/wiki/Convolution
- https://www.khanacademy.org/math/integral-calculus/indefinite-definite-integrals

Questions

- Ask me
- Mail me at mftspirit@gmail.com