

1 Kalman Filter

Lets suppose there a data generative process is a linear of its past values, i.e. the current process parameters are a linear combination of previous process parameters. In this respect we can calculate the process parameters by minimising the least squared error between the prediction and the true values, $\omega = (\mathbf{Y}^t \mathbf{Y})^{-1} \mathbf{Y}^t \mathbf{f}$, where \mathbf{Y} is the input data and \mathbf{f} is the target. But this method requires calculating a new \mathbf{Y}^{-1} (a $O(n^3)$ operation) when we receive any new data. If the process is generating a time series data (getting data one at a time), then online updating the parameters can be incredibly costly.

However, it can be shown that we can calculate new parameters without calculating a new inverse, $\omega_n = \omega_{n-1} + L \times (f_n - \omega_{n-1}^t y_n)$. It means that we can calculate new parameters by updating the old parameters with the scaled prediction error on new data using the old parameters. Of course we need to assume some initial values for our parameters. But, we still do not account for any noise in our process.

In a Kalman Filter, we may assume a process $\omega_n = \mathbf{A}\omega_{n-1} + \epsilon(n)$, where the process noise $\epsilon \sim \mathcal{N}(0, \mathbf{Q})$. Here we can assume that our parameters are a linear transformation of previous parameters corrupted by Gaussian noise. Similarly, we can assume that our observation $f_n = \omega_n y_n + \xi(n)$, where observation noise $\xi \sim \mathcal{N}(0, R)$.

[1] claims that we can predict the monthly S&P 500 index using a Kalman filter. In order to verify his claim, I have built Algorithm 1 -

Algorithm 1 S&P 500 Index Prediction using a Kalman Filter

```

procedure KALMAN( $\mathbf{s}, o, \alpha$ )                                ▷  $\mathbf{s}$  = index,  $o$  = order,  $\alpha$  = is a constant

     $N = \text{size}(\mathbf{s}, 1);$                                        ▷ Number of months
     $\mathbf{Y} = \text{zeros}(N - o + 1, o);$                                ▷ Input data in  $o$  order windows
     $\mathbf{W} = \text{zeros}(N - o + 1, o);$                                ▷ Weights at every point
     $\mathbf{e} = \text{zeros}(N - o + 1, 1);$                                ▷ Prediction error at every point
     $\mathbf{K} = \text{zeros}(N - o + 1, o);$                                ▷ Kalman gain at every point

     $\mathbf{A} = \mathbf{I}_o$                                                ▷  $\mathbf{I}_o$  is the  $o$  order Identity
     $\mathbf{Q} = \alpha \mathbf{I}_o$                                          ▷ Process noise covariance

     $\mathbf{P} = \text{ones}(o)$                                            ▷ Initial value of  $\mathbf{P}$ 
     $\mathbf{W}(1, :) = o^{-1} \text{ones}(1, o)$ 

    for  $i \leftarrow 1, N - o$  do
         $\mathbf{Y}(n, :) = \mathbf{s}(i : i + o - 1)$                        ▷ Input is last  $o$  months' index

         $\hat{\mathbf{w}} = \mathbf{A}\mathbf{W}(n - 1, :)^t$                                ▷ Prediction
         $\hat{\mathbf{P}} = \mathbf{A}\mathbf{P}\mathbf{A}^t + \mathbf{Q}$ 

         $\mathbf{e}(n) = \mathbf{s}(n + o - 1) - \mathbf{Y}(n, :)\hat{\mathbf{w}}$                ▷ Prediction Error
         $\mathbf{K}(n, :) = (\hat{\mathbf{P}}\mathbf{Y}(n, :)^t) / (R + \mathbf{Y}(n, :)\hat{\mathbf{P}}\mathbf{Y}(n, :)^t)$    ▷ Kalman Gain

         $\mathbf{W}(n, :) = \mathbf{W}(n - 1, :) + \mathbf{K}(n, :)\mathbf{e}(n)$            ▷ Correction
         $\mathbf{P} = (\mathbf{I}_o - \mathbf{K}(n, :)^t \mathbf{Y}(n, :))\mathbf{P}$ 

```

References

- [1] N. Mahler, "Modeling the S & P 500 index using the Kalman filter and the LagLasso," in *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on, Sept 2009*, pp. 16.