The filters: theory and computation

Muhammad Yaseen

2026-11-05

Table of contents

1 Abstract **1**

2 The HP filter **2**

3 Choosing lambda from the data **2**
3.1 The criterion of the modified HP filter 2
3.2 The two criteria on the US data 2

4 The end-point bias **4**
4.1 Weights 4
4.2 Gains 5
4.3 Loss 6

5 The fully modified HP filter **7**
5.1 End-point schemes 7
5.2 Two forms of the weights 7
5.3 The procedure 8

6 References **9**

1 Abstract

This vignette sets out what the three filters compute: the HP filter as a penalised least-squares problem, the criterion of the modified HP filter and the exact cross-validation it approximates, the weights, gains and end-point loss that measure the end-point bias, and the end-point schemes of the fully modified HP filter. It also says how the package computes them, in $O(T)$ operations per smoothing parameter.

2 The HP filter

With the series $\mathbf{x} = (x_1, \dots, x_T)^\top$ and the $(T-2) \times T$ matrix $\mathbf{K}$ of second differences (rows $(1, -2, 1)$), the trend minimises $\|\mathbf{x} - \mathbf{g}\|^2 + \lambda \|\mathbf{K}\mathbf{g}\|^2$, so

$$\hat{\mathbf{g}} = (\mathbf{I} + \lambda \mathbf{A})^{-1} \mathbf{x} = \mathbf{B}\mathbf{x}, \qquad \mathbf{A} = \mathbf{K}^\top \mathbf{K},$$

and $\hat{\mathbf{c}} = \mathbf{x} - \hat{\mathbf{g}}$. Because $\mathbf{K}\mathbf{1} = \mathbf{0}$, every row of $\mathbf{B}$ sums to one: the trend is a weighted average of the observations.

Computation. $\mathbf{I} + \lambda \mathbf{A}$ is symmetric, positive definite and pentadiagonal. The package factors it as $\mathbf{L}\mathbf{D}\mathbf{L}^\top$ with a unit lower-triangular $\mathbf{L}$ of bandwidth two and solves in $O(T)$ operations; version 0.1.0 solved a dense $T \times T$ system, $O(T^3)$, for every λ of the search.

3 Choosing lambda from the data

3.1 The criterion of the modified HP filter

McDermott (1997) chooses λ by cross-validation: leave out one observation, predict it from the trend of the others, and sum the squared errors. Generalised cross-validation (Craven and Wahba 1979) replaces the leave-one-out errors by

$$\mathrm{GCV}(\lambda) = \frac{T^{-1} \sum_t \left(x_t - \hat{g}_t(\lambda)\right)^2}{\left(1 - T^{-1} \operatorname{tr} \mathbf{B}(\lambda)\right)^2}$$

(Choudhary et al. 2014, eq. 5), and the papers and their code use the approximation

$$\mathrm{GCV}(\lambda) \approx \frac{1}{T} \left(1 + \frac{2T}{\lambda}\right) \sum_t \left(x_t - \hat{g}_t(\lambda)\right)^2$$

(their eq. 6, attributed to Silverman 1984). `mhp_filter()` minimises the approximation over $\lambda = 1, \dots,$ `max_lambda` by default, and the exact criterion with `criterion = "exact"`: the eigenvalues μ_i of $\mathbf{A}$, computed once, give $\operatorname{tr} \mathbf{B}(\lambda) = \sum_i (1 + \lambda \mu_i)^{-1}$ and the residual sum of squares for every λ in $O(T)$ operations.

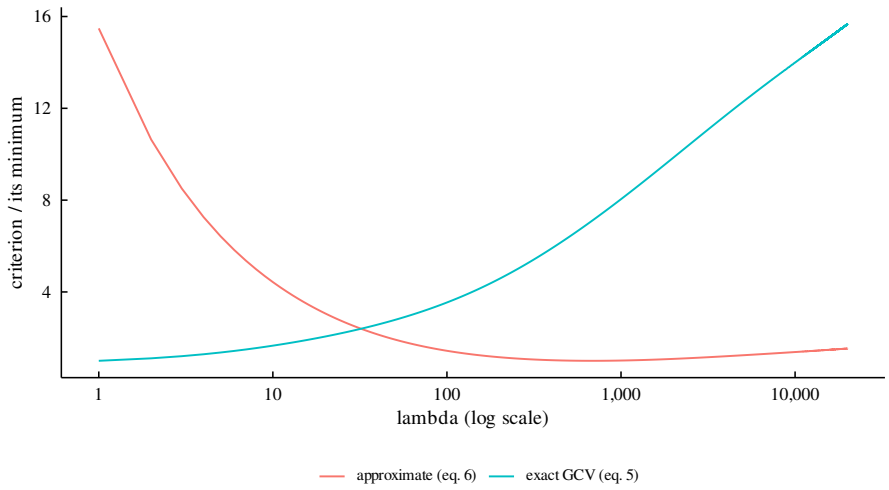
3.2 The two criteria on the US data

```
data(us_macro)
x <- log(us_macro$gdp)
ap <-
  mhp_filter(
      x          = x
    , max_lambda = 20000
    , as_dt      = FALSE
    , log_scale  = FALSE
```

```
    , criterion  = "approximate"
    )
ex <- suppressWarnings(
  mhp_filter(
      x          = x
    , max_lambda = 20000
    , as_dt      = FALSE
    , log_scale  = FALSE
    , criterion  = "exact"
    )
  )
c(approximate = ap$lambda, exact = ex$lambda)
```

```
approximate       exact
        683           1
```

```
d <- data.table::data.table(
    lambda    = rep(seq_len(length.out = 20000), 2)
  , criterion = c(ap$gcv_path / min(ap$gcv_path), ex$gcv_path /
↪  min(ex$gcv_path))
  , type      = rep(c("approximate (eq. 6)", "exact GCV (eq. 5)"), each =
↪  20000)
  )
ggplot2::ggplot(data = d, mapping = ggplot2::aes(x = lambda, y =
↪  criterion, colour = type)) +
  ggplot2::geom_line() +
  ggplot2::scale_x_log10(breaks = scales::breaks_log(n = 6), labels =
↪  scales::label_comma()) +
  ggplot2::labs(x = "lambda (log scale)", y = "criterion / its minimum",
↪  colour = NULL) +
  mhp_theme(
      base_size       = 11
    , base_family     = "serif"
    , ticks_inward    = TRUE
    , legend_position = "bottom"
    )
```



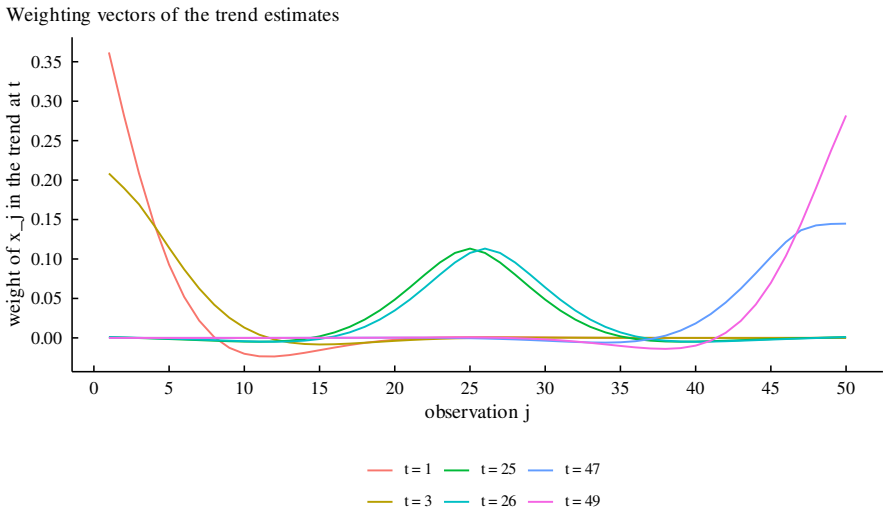
The exact criterion falls steadily as λ falls, down to $\lambda = 1$, where the trend almost interpolates the series: generalised cross-validation is known to undersmooth series whose deviations from the trend are autocorrelated, as business cycles are. The approximation adds a much heavier penalty for small λ (at $\lambda = 683$ the factor $1 + 2T/\lambda$ is 1.59, against 1.17 for the exact one), and that penalty, not cross-validation as such, gives the modified HP filter its interior minimum. The package keeps the published approximation as the default.

4 The end-point bias

4.1 Weights

Row t of $\mathbf{B}$ holds the weights of $\hat{g}_t$. In the middle of the series they are symmetric; near the ends they are not, and the end observation gets a large weight in its own trend (Hanif et al. 2017, Figure 1).

```
w <-
  hp_weights(
      x      = 50
    , lambda = 100
    , form   = "rows"
    , rows   = NULL
    )
plot(x = w)
```



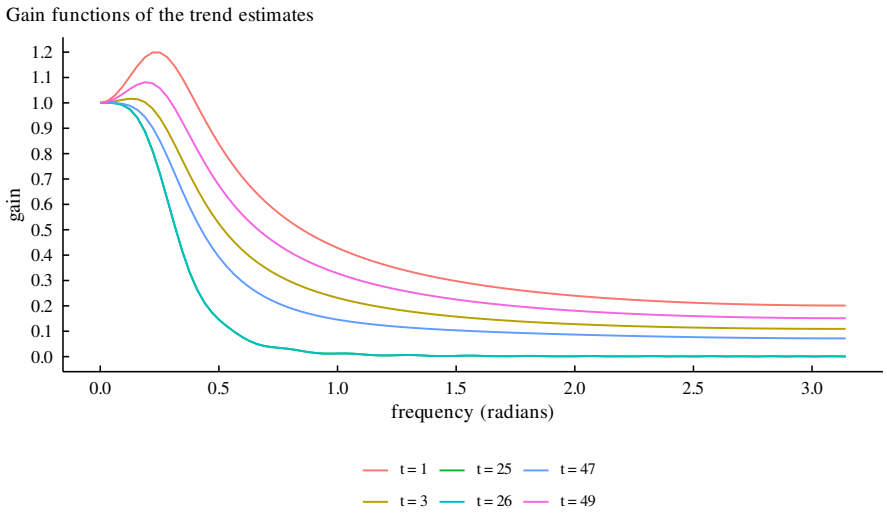
4.2 Gains

The gain of $\hat{g}_t$ at frequency ω,

$$P_t(\omega) = \left| \sum_{j=1}^{T} b_{tj}\, e^{i\omega(j-t)} \right|,$$

is the factor by which the trend scales an oscillation of that frequency (eq. 6 of Hanif et al. 2017). In the middle the gain falls to zero at high frequencies; at the ends it does not, so the trend keeps part of the cycle.

```
g <-
  hp_gain(
      x      = 50
    , lambda = 100
    , form   = "rows"
    , theta  = seq(from = 0, to = pi, length.out = 101)
    , rows   = NULL
    )
plot(x = g)
```



The package computes $\sum_j b_{tj} e^{i\omega j}$ for all t at once as $\mathbf{B}\cos(\omega\cdot)$ and $\mathbf{B}\sin(\omega\cdot)$: two solves per frequency, without forming $\mathbf{B}$.

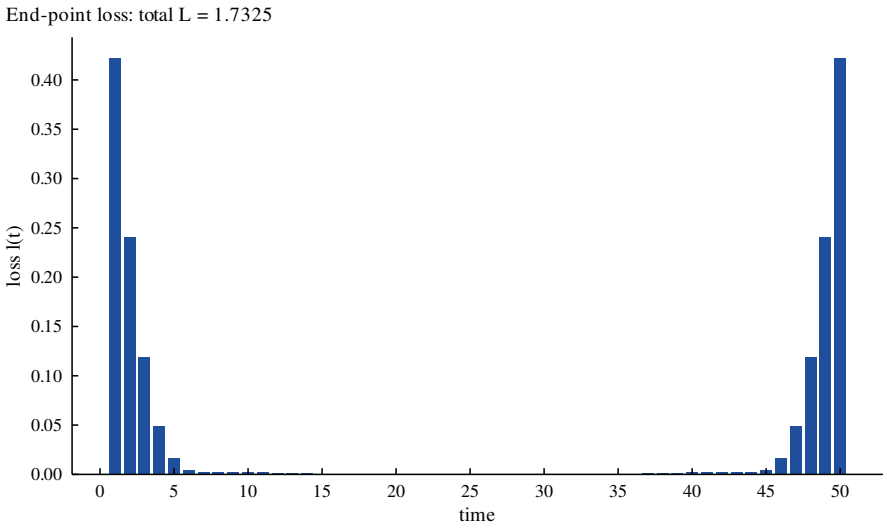
4.3 Loss

Bloechl (2014) measures the distortion by the distance of each gain from the gain at the centre c of the series,

$$l(t) = \theta \sum_{\omega} \left[P_c(\omega) - P_t(\omega)\right]^2, \qquad L = \sum_{t=1}^{T} l(t),$$

with $\omega = 0, 0.1, \ldots, 3.1$ and $\theta = 0.1$ (eqs. 7 and 8 of Hanif et al. 2017). It depends on T and the smoothing parameters only.

```
hl <-
  hp_loss(
      x      = 50
    , lambda = 100
    , form   = "rows"
    , theta  = seq(from = 0, to = 3.1, by = 0.1)
    , weight = 0.1
    )
plot(x = hl)
```



5 The fully modified HP filter

5.1 End-point schemes

Hanif et al. (2017) raise the smoothing parameter of the k values at each end: $\lambda_t = \lambda$ in the middle and $\lambda + \alpha j^i$ for the j-th value from the middle, linear ($i = 1$) or quadratic ($i = 2$).

```
hp_schedule(
    T      = 50
  , lambda = 100
  , alpha  = 10
  , k      = 20
  , power  = 1
  , form   = "rows"
  )
```

```
 [1] 300 290 280 270 260 250 240 230 220 210 200 190 180 170 160 150 140 130 120
[20] 110 100 100 100 100 100 100 100 100 100 100 100 110 120 130 140 150 160 170 180
[39] 190 200 210 220 230 240 250 260 270 280 290 300
```

5.2 Two forms of the weights

The published MATLAB code multiplies row t of $\mathbf{A}$ by λ_t:

$$\hat{\mathbf{g}} = (\mathbf{I} + \mathrm{diag}(\lambda_t)\mathbf{A})^{-1}\mathbf{x} \qquad (\texttt{form = "rows"}, \text{ the default}).$$

This matrix is not symmetric, but $\mathbf{I} + \mathrm{diag}(\lambda_t)\mathbf{A} = \mathrm{diag}(\lambda_t)(\mathrm{diag}(1/\lambda_t) + \mathbf{A})$, so the package solves the symmetric pentadiagonal system $(\mathrm{diag}(1/\lambda_t) + \mathbf{A})\hat{\mathbf{g}} =$

$\mathrm{diag}(1/\lambda_t)\mathbf{x}$. The penalised least-squares problem with one smoothing parameter per second difference, $\|\mathbf{x}-\mathbf{g}\|^2 + \sum_r \lambda_r (\mathbf{K}\mathbf{g})_r^2$, gives instead

$$\hat{\mathbf{g}} = (\mathbf{I} + \mathbf{K}^\top \mathrm{diag}(\lambda_r)\mathbf{K})^{-1}\mathbf{x} \qquad (\texttt{form = "differences"}),$$

the form in which Bloechl (2014) and eq. 9 of the paper write the scheme. Both are $O(T)$; for a constant λ they coincide.

```
y <- us_macro$gdp
rows <-
  fmhp_filter(
      x               = y
    , max_lambda      = 3000
    , k_linear        = 15
    , k_quadratic     = NULL
    , alpha_linear    = seq(from = 0, to = 2000, by = 10)
    , alpha_quadratic = seq(from = 0, to = 2000, by = 5)
    , schemes         = c("linear", "quadratic")
    , form            = "rows"
    , lambda          = NULL
    , log_scale       = TRUE
    , as_dt           = FALSE
    )
diffs <-
  fmhp_filter(
      x               = y
    , max_lambda      = 3000
    , k_linear        = 15
    , k_quadratic     = NULL
    , alpha_linear    = seq(from = 0, to = 2000, by = 10)
    , alpha_quadratic = seq(from = 0, to = 2000, by = 5)
    , schemes         = c("linear", "quadratic")
    , form            = "differences"
    , lambda          = NULL
    , log_scale       = TRUE
    , as_dt           = FALSE
    )
rbind(
    rows        = rows$schemes[scheme == rows$scheme]
  , differences = diffs$schemes[scheme == diffs$scheme]
  )
```

```
      scheme  k alpha      loss lambda          gcv
1: quadratic 22    15 0.6163442   1448 0.0003335623
2: quadratic 22    95 1.1645828   1410 0.0003353531
```

5.3 The procedure

1. λ^{MHP} of the modified HP filter.

2. For the linear and the quadratic scheme, the weight α of a grid that minimises L at λ^{MHP}.
3. With that α, λ^{FMHP} minimising the criterion of the modified HP filter again.
4. The scheme with the smaller loss in step 2 gives the trend and the cycle.

The defaults are those of the MATLAB code: $\lambda \leq 3000$; $k = 15$ and $\alpha \in \{0, 10, \ldots, 2000\}$ for the linear scheme; $k = \mathrm{round}(\min(22, T/2 - 5))$ and $\alpha \in \{0, 5, \ldots, 2000\}$ for the quadratic one; logarithms. The text of the paper states $k = 20$; `k_linear` and `k_quadratic` set it. MATLAB rounds halves up, and so does the package (the centre c of the loss and the quadratic k); vignette 4 shows that this reproduces the published losses.

6 References

Bloechl, A. (2014). Reducing the excess variability of the Hodrick-Prescott filter by flexible penalization. *Discussion Paper* 2014-1, University of Munich.

Choudhary, M. A., Hanif, M. N. and Iqbal, J. (2014). On smoothing macroeconomic time series using the modified HP filter. *Applied Economics*, 46(19), 2205–2214.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. *Numerische Mathematik*, 31, 377–403.

Hanif, M. N., Iqbal, J. and Choudhary, M. A. (2017). Fully modified HP filter. *SBP Working Paper Series*, No. 88, State Bank of Pakistan.

McDermott, C. J. (1997). An automatic method for choosing the smoothing parameter in the HP filter. Unpublished, International Monetary Fund.

Silverman, B. W. (1984). A fast and efficient cross-validation method for smoothing parameter choice in spline regression. *Journal of the American Statistical Association*, 79(387), 584–589.