Validation and speed

Muhammad Yaseen

2026-11-05

Table of contents

1 Abstract **1**

2 Against version 0.1.0 **1**

3 Against dense computations **2**

4 Against the MATLAB code **3**

5 Against the published values **3**

6 Speed **4**

7 References **4**

1 Abstract

This vignette sets out how the results of the package are checked: against version 0.1.0, against direct dense computations, against the MATLAB code of the authors and against the published values. It ends with timings of this version against version 0.1.0.

2 Against version 0.1.0

The results of every exported function of version 0.1.0 were saved before the rewrite (`dev/legacy`), and the test `test-legacy.R` compares the new results with them: the smoothing parameters are identical, and the trends, cycles and criteria agree to 10^{-8}. The names of the functions, their arguments and the components of their results are kept; new arguments come after the old ones.

3 Against dense computations

The package never forms the $T \times T$ smoothing matrix. Forming it shows that the banded solves give the same trend:

```
x <- log(us_macro$gdp)
T <- length(x)
K <- diff(diag(T), differences = 2)
A <- crossprod(K)
fit <-
  hp_filter(
      x         = x
    , lambda    = 1600
    , as_dt     = FALSE
    , log_scale = FALSE
    )
max(abs(solve(a = diag(T) + 1600 * A, b = x) - fit$trend))
```

```
[1] 9.425349e-12
```

The same holds for the fully modified filter, with the end-point weights it chose, in both forms:

```
for (form in c("rows", "differences")) {
  f <-
    fmhp_filter(
        x               = x
      , max_lambda      = 3000L
      , k_linear        = 15L
      , 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            = form
      , lambda          = NULL
      , log_scale       = FALSE
      , as_dt           = FALSE
      )
  M <- if (form == "rows") diag(T) + diag(f$lambda_t) %*% A else diag(T)
 ↪  + t(K) %*% diag(f$lambda_t) %*% K
  cat(form, ":", max(abs(solve(a = M, b = x) - f$trend)), "\n")
}
```

```
rows : 1.094769e-11
differences : 1.046665e-10
```

and the exact cross-validation computed from the eigenvalues of $\mathbf{A}$ agrees with the trace of the dense smoothing matrix:

```
lam <- 683
B <- solve(diag(T) + lam * A)
```

```r
gcv_dense <- mean((x - B %*% x)^2) / (1 - sum(diag(B)) / T)^2
ex <- suppressWarnings(
  mhp_filter(
      x          = x
    , max_lambda = 2000
    , as_dt      = FALSE
    , log_scale  = FALSE
    , criterion  = "exact"
    )
  )
c(dense = gcv_dense, package = ex$gcv_path[lam])
```

```
       dense      package
0.0002064307 0.0002064307
```

4 Against the MATLAB code

The tests carry line-by-line ports of the MATLAB functions of Choudhary et al. (2014) and Hanif et al. (2017), with dense matrices and explicit loops (`tests/testthat/helper-matlab.R`). `mhp_filter()` returns the same smoothing parameter as the first on 80 quarters of US GDP, with trend and cycle to 10^{-9}; `fmhp_filter()` returns the same smoothing parameters and weights as the second on a simulated series, with losses to 10^{-10} and trend to 10^{-8}.

5 Against the published values

```r
ref <-
  mhp_reference(
      source   = NULL
    , evaluate = TRUE
    , level    = 1
    )
ref[, .(id, value, computed = round(x = computed, digits = 3), agrees)]
```

```
                          id value computed agrees
 1:    hanif2017_t3a_hp_T55 1.735    1.735   TRUE
 2:    hanif2017_t3a_hp_T54 1.733    1.733   TRUE
 3:    hanif2017_t3a_hp_T40 1.731    1.731   TRUE
 4:    hanif2017_t3b_hp_T72 1.757    1.757   TRUE
 5:   hanif2017_t3b_hp_T103 1.770    1.770   TRUE
 6:  hanif2017_t3a_ours_T55 0.626    0.626   TRUE
 7:  hanif2017_t3a_ours_T54 0.627    0.627   TRUE
 8:  hanif2017_t3a_ours_T50 0.623    0.623   TRUE
 9: hanif2017_fig1_hp_range 0.385    0.385   TRUE
10:   hanif2017_fig1a_range 0.182    0.182   TRUE
```

```
11:   hanif2017_t3b_ours_US 0.797    1.000  FALSE
```

The one row that does not agree is annotated in the file and discussed in the vignette "Reproducing the fully modified HP filter"; the test of published values skips annotated rows.

6 Speed

Version 0.1.0 solved a dense $T \times T$ system for every λ of the search, $O(T^3)$ operations each. This version factors the pentadiagonal system in $O(T)$, and in the search updates the bands in place. The timings below were taken by `tools/bench.R` on simulated series (median of repeated runs, milliseconds) and are stored with the package:

```
b <- data.table::fread(system.file("extdata", "bench.csv", package =
 ↪  "mhpfilter"))
w <- data.table::dcast(data = b, formula = case ~ version, value.var =
 ↪  "median_ms")
v <- sort(unique(b$version))
w[, speedup := round(get(v[1]) / get(v[2]))]
w
```

```
           case       0.1.0       1.0.0 speedup
1:     batch_10 1661.892177  435.077582       4
2:      hp_T200    1.995088    0.714376       3
3:     mhp_T200  165.069662   38.895816       4
4: mhp_T200_big 6370.139562 1318.158352       5
5:     mhp_T500 2133.439273  106.096296      20
```

The cases are `hp_filter()` and `mhp_filter()` with 200 and 500 observations, the search up to $\lambda = 100000$ with 200 observations, and `mhp_batch()` with ten series of 200 observations. The gain grows with the length of the series, since the old cost per λ grew as T^3 and the new one grows as T.

7 References

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.

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