**Comparison**

library(bigstatsr)

library(tidyverse)

**Data**

# Create two matrices, one with some structure, one without

n <- 20e3

seq\_m <- c(1e3, 3e3, 10e3)

sizes <- seq\_along(seq\_m)

X <- E <- list()

for (i in sizes) {

m <- seq\_m[i]

U <- matrix(0, n, 10); U[] <- rnorm(length(U))

V <- matrix(0, m, 10); V[] <- rnorm(length(V))

E[[i]] <- matrix(rnorm(n \* m), n, m)

X[[i]] <- tcrossprod(U, V) + E[[i]]

}

I use matrices of different sizes. Some are structured with 10 hidden components, and some with only random data.

**Optimized math library**

I linked my R installation with OpenBLAS, an optimized parallel matrix library.

(NCORES <- RhpcBLASctl::get\_num\_cores())

## [1] 6

RhpcBLASctl::blas\_set\_num\_threads(NCORES)

## detected function goto\_set\_num\_threads

**Compared methods**

models <- tribble(

~method, ~fun, ~params,

"bigstatsr - 1 core", bigstatsr::big\_randomSVD, list(k = 10),

"bigstatsr - 6 cores", bigstatsr::big\_randomSVD, list(k = 10, ncores = NCORES),

"Rspectra", RSpectra::svds, list(k = 10),

"irlba", irlba::irlba, list(nv = 10, nu = 10),

"svd", svd::propack.svd, list(neig = 10),

"rsvd", rsvd::rsvd, list(k = 10)

) %>%

mutate(size = list(sizes), structured = list(c(TRUE, FALSE))) %>%

unnest(size, .drop = FALSE) %>%

unnest(structured, .drop = FALSE) %>%

mutate(user\_time = NA, real\_time = NA, pcs = list(NA))

**Computing**

# Filling this data frame with times and PC scores for each method and dataset

for (i in rows\_along(models)) {

mat <- `if`(models$structured[[i]], X, E)[[models$size[[i]]]]

time <- system.time({

if (grepl("bigstatsr", models$method[[i]])) mat <- as\_FBM(mat)

res <- do.call(models$fun[[i]], args = c(list(mat), models$params[[i]]))

})

models[["user\_time"]][[i]] <- time[1]

models[["real\_time"]][[i]] <- time[3]

models[["pcs"]][[i]] <- res

}

## Warning in (function (X, neig = min(m, n), opts = list()) : Only 4 singular triplets converged

## within 50 iterations.

## Warning in (function (X, neig = min(m, n), opts = list()) : Only 5 singular triplets converged

## within 50 iterations.

## Warning in (function (X, neig = min(m, n), opts = list()) : Only 5 singular triplets converged

## within 50 iterations.

models <- mutate(models, size = seq\_m[size])

**Timings**

models %>%

ggplot(aes(size / 1000, real\_time, color = method)) +

theme\_bigstatsr() +

geom\_point(cex = 6) +

geom\_line(aes(linetype = method), lwd = 2) +

facet\_grid(structured ~ ., scales = "free") +

theme(legend.position = c(0.25, 0.87),

legend.key.width = unit(6, "line")) +

labs(x = sprintf("ncol (x1000) (nrow = %d)", n), y = "Time (in seconds)",

color = "Methods:", linetype = "Methods:")

models %>%

filter(size == max(seq\_m)) %>%

select(method, structured, user\_time, real\_time)

## # A tibble: 12 x 4

## method structured user\_time real\_time

##

## 1 bigstatsr - 1 core TRUE 8.10 8.68

## 2 bigstatsr - 1 core FALSE 106. 107.

## 3 bigstatsr - 6 cores TRUE 0.456 6.80

## 4 bigstatsr - 6 cores FALSE 0.616 45.3

## 5 Rspectra TRUE 17.9 3.39

## 6 Rspectra FALSE 329. 56.1

## 7 irlba TRUE 16.3 3.09

## 8 irlba FALSE 399. 68.8

## 9 svd TRUE 34.2 6.11

## 10 svd FALSE 274. 46.9

## 11 rsvd TRUE 4.12 3.89

## 12 rsvd FALSE 4.06 3.88

**Errors**

true1 <- svd(X[[1]], nu = 10, nv = 10)

true2 <- svd(E[[1]], nu = 10, nv = 10)

bdiff <- function(x, y) {

if (ncol(x) < ncol(y)) return(Inf)

s = sign(x[1, ] / y[1, ])

max(apply(sweep(x, 2, s, '\*') - y, 2, crossprod))

}

models %>%

filter(size == min(seq\_m)) %>%

mutate(error = map2\_dbl(structured, pcs, ~{

true <- `if`(.x, true1, true2)

bdiff(.y$u, true$u)

})) %>%

select(method, structured, error)

## # A tibble: 12 x 3

## method structured error

##

## 1 bigstatsr - 1 core TRUE 1.63e-27

## 2 bigstatsr - 1 core FALSE 1.08e- 5

## 3 bigstatsr - 6 cores TRUE 1.29e-27

## 4 bigstatsr - 6 cores FALSE 1.08e- 5

## 5 Rspectra TRUE 1.63e-27

## 6 Rspectra FALSE 3.07e-18

## 7 irlba TRUE 9.06e-26

## 8 irlba FALSE 1.78e- 7

## 9 svd TRUE 6.83e-27

## 10 svd FALSE Inf

## 11 rsvd TRUE 7.02e-13

## 12 rsvd FALSE 2.38e+ 0

**Conclusion**

* Packages {rsvd} and {svd} don't give results precise enough when data is not structured.
* Packages {bigstatsr} and {irlba} are less precise (but precise enough!) than {RSpectra} because of a different tolerance parameter they use.
* Package {bigstatsr} is as fast as the other packages while not relying on matrix operations (see user timings above). So, even if you don't have your R installation linked to some optimized math library, you would get the same performance. On the contrary, the other methods would likely be much slower if not using such optimized library.