comparison_LinearRegression.r

max

2021-03-13

```
library(npowerPrioR)
```

```
## Loading required package: parallel
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
## Loading required package: rstan
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
## Loading required package: bridgesampling
```

```
source("../Linear_Regression/data_regression_NIG_scenario_1.r")

lm.data <- list(
  N0 = N_0,
  P = P,
  X0 = X_0,
  y0 = y_0,
  mu_beta = rep(0, P),
  lambda_0 = solve(vb * diag(P)),
  alpha0 = as,
  beta0 = bs,
  a_0 = NULL
)
invlambda0 <- solve(lm.data$lambda_0)
###
get_mal_NIG_regression <- function(y0, X0, n0, mu0, lambda0, alpha0, beta0, a_0){
  P <- ncol(X0)
  if(length(mu0) != P) stop("mu0 is not the same dimension as X")
  Xstar <- sqrt(a_0) * X0
  ystar <- sqrt(a_0) * y0
  lambda_n <- t(Xstar)%*%Xstar + lambda0
  mu_n <- solve(lambda_n) %*% (lambda0%*%mu0 + t(Xstar)%*%ystar)
  alpha_n <-  as + (n0*a_0)/2
```

```
  beta_n <- beta0 + .5 * ( t(ystar)%*%ystar + t(mu0)%*%lambda0%*%mu0 - t(mu_n)%*%lambda_n%*%mu_n )
  det0 <- det(lambda0)
  det1 <- det(lambda_n)
  ans <- -(a_0 * n0/2) * log(2*pi) + .5 * (log(det0) - log(det1)) + (alpha0 * log(beta0) - alpha_n * log
  return(ans)
}
#
l_a0 <- function(x) {
  get_mal_NIG_regression(
    y0 = lm.data$y0,
    X0 = lm.data$X0,
    n0 = lm.data$N0,
    mu0 = lm.data$mu_beta,
    lambda0 = invlambda0,
    alpha0 = lm.data$alpha0,
    beta0 = lm.data$beta0,
    a_0 = x
  )
}
l_a0 <- Vectorize(l_a0)

########

maxA <- 1
prior <- stan_model("../Linear_Regression/stan/simple_linear_regression_NIG_prior.stan")
```

```
## Trying to compile a simple C file
```

```
## Running /usr/local/lib/R/bin/R CMD SHLIB foo.c
## gcc -I"/usr/local/lib/R/include" -DNDEBUG   -I"/home/max/R/x86_64-pc-linux-gnu-library/4.0/Rcpp/inclu
## In file included from /home/max/R/x86_64-pc-linux-gnu-library/4.0/RcppEigen/include/Eigen/Core:88,
##                  from /home/max/R/x86_64-pc-linux-gnu-library/4.0/RcppEigen/include/Eigen/Dense:1,
##                  from /home/max/R/x86_64-pc-linux-gnu-library/4.0/StanHeaders/include/stan/math/prim/
##                  from <command-line>:
## /home/max/R/x86_64-pc-linux-gnu-library/4.0/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: err
##   613 | namespace Eigen {
##       | ^~~~~~~~~
## /home/max/R/x86_64-pc-linux-gnu-library/4.0/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:17: er
##   613 | namespace Eigen {
##       |                 ^
## In file included from /home/max/R/x86_64-pc-linux-gnu-library/4.0/RcppEigen/include/Eigen/Dense:1,
##                  from /home/max/R/x86_64-pc-linux-gnu-library/4.0/StanHeaders/include/stan/math/prim/
##                  from <command-line>:
## /home/max/R/x86_64-pc-linux-gnu-library/4.0/RcppEigen/include/Eigen/Core:96:10: fatal error: complex:
##    96 | #include <complex>
##       |          ^~~~~~~~~
## compilation terminated.
## make: *** [/usr/local/lib/R/etc/Makeconf:172: foo.o] Error 1
```

```
# direct method
J <- 20
epsilon <- 0.05

adaptive.time <- system.time(
  adaptive.ca0.estimates <- build_grid(compiled.model.prior = prior, eps = epsilon,
```

```
                                        M = maxA, J = J, v1 = 10, v2 = 10,
                                        stan.list = lm.data, pars = c("beta", "sigma_sq"))
)
```

```
## Warning: There were 94 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: There were 6 transitions after warmup that exceeded the maximum treedepth. Increase max_tree
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: There were 2 chains where the estimated Bayesian Fraction of Missing Information was low. Se
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: The largest R-hat is 1.05, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
# VR2018
Delta.a <- 0.01
a0s.vr2018 <- seq(0, maxA, by = Delta.a)

vr2018.time <- system.time(
  vr2018.estimates <-  create_lc_df_derivOnly(a0_grid = a0s.vr2018,
                              compiled.model.prior = prior,
                              stan.list = lm.data, pars = c("beta", "sigma_sq") )
)
```

```
## Warning: There were 19 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. Se
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: The largest R-hat is 1.07, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quanti
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
## Warning: There were 2 transitions after warmup that exceeded the maximum treedepth. Increase max_tree
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: There were 1 chains where the estimated Bayesian Fraction of Missing Information was low. S
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
write.csv(vr2018.estimates$result,
          file = "Gaussian_VR2018.csv", row.names = FALSE)
adaptive.time
```

```
##    user  system elapsed
##  23.146   0.104  23.343
```

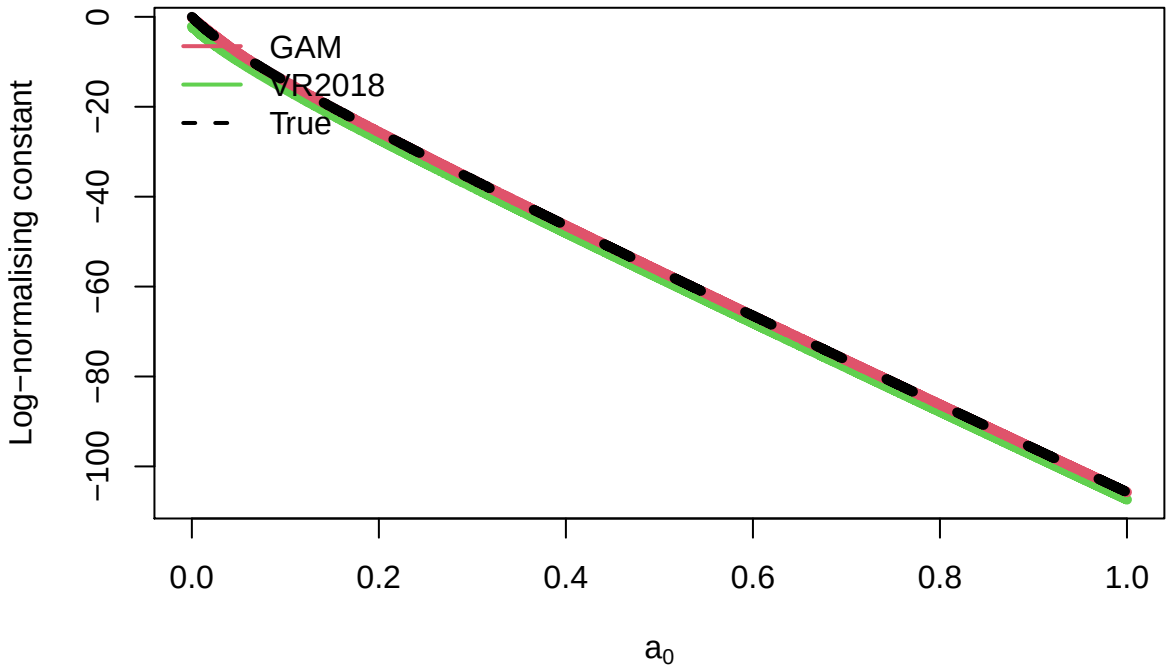
```
vr2018.time
```

```
##    user  system elapsed
##  52.773   0.060  53.134
```

```
###
## Now the approximations
adapt.gam <-  mgcv::gam(lc_a0 ~ s(a0, k = J), data = adaptive.ca0.estimates$result)
vr2018.estimates$result$la0_est <- cumsum(vr2018.estimates$result$deriv_lc) * Delta.a


## Finally, comparisons

K <- 20000
pred.a0s <- seq(0, maxA, length.out = K)

true.la0s <- l_a0(pred.a0s)

adaptive.preds <- predict(adapt.gam, newdata = data.frame(a0 = pred.a0s))

vr2018.preds <- approx(x = vr2018.estimates$result$a0,
                       y =  vr2018.estimates$result$la0_est,
                       xout =  pred.a0s)

plot(vr2018.preds, type = "l", lwd = 5,
     col = 3,
     xlab = expression(a[0]), ylab = "Log-normalising constant")
lines(pred.a0s, adaptive.preds, col = 2, lwd = 5)
lines(pred.a0s, true.la0s, lwd = 5, lty = 2, add = TRUE)
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "add" is not a graphical
## parameter
```

```
legend(x = "topleft", legend =  c("GAM", "VR2018", "True"),
       col = c(2, 3, 1), lwd  = 2, lty = c(1, 1, 2), bty = 'n')
```




```
preds.list <- list(
  adaptive = adaptive.preds,
  VR2018 = vr2018.preds$y
)

ntrue.la0s <- true.la0s

lapply(preds.list, function(pred) sqrt(mean( ( pred- ntrue.la0s)^2 )) )
```

```
## $adaptive
## [1] 0.03665582
##
## $VR2018
## [1] 1.690613
```

```
lapply(preds.list, function(pred) mean( abs( pred- ntrue.la0s) ))
```

```
## $adaptive
## [1] 0.01402521
##
## $VR2018
## [1] 1.689064
```