project

Amanpreet Singh Saini

2024-11-27

```
rm(list =ls())
library(R2OpenBUGS)
```

```
## Warning: package 'R2OpenBUGS' was built under R version 4.4.2
```

```
fully_bayesian_posterior <- function(Y, k, n, alpha_prior, beta_prior) {
  # Define BUGS model
  model_code <- "
  model {
    for (i in 1:k) {
      Y[i] ~ dbin(theta[i], n)  # Binomial likelihood
      theta[i] ~ dbeta(a, b)    # Beta prior
    }
    a ~ dunif(0, 1000)          # Prior for alpha
    b ~ dunif(0, 1000)          # Prior for beta
  }"

  writeLines(model_code, "model.txt")
  # data
  data_list <- list(Y = Y, k = k, n = n)
  inits <- function() list(a = runif(1, 0, 5), b = runif(1, 0, 5))

  # Parameters
  parameters <- c("a", "b")

  # Run BUGS and measure time
  start_time <- Sys.time()
  bugs_out <- bugs(
    data = data_list, inits = inits, parameters.to.save = parameters,
    model.file = "model.txt", n.chains = 3, n.iter = 5000, n.burnin = 1000, n.thin = 2,
    DIC = TRUE, debug = TRUE  # Enable debugging
  )
  end_time <- Sys.time()

  # Time taken
  computation_time <- difftime(end_time, start_time, units = "secs")

  # Extract posterior mean
  if (!is.null(bugs_out$sims.list)) {
    posterior_alpha <- mean(bugs_out$sims.list$a)
    posterior_beta <- mean(bugs_out$sims.list$b)
```

```
  } else {
    cat("BUGS did not run correctly. Check your setup.\n")
    return(NULL)
  }

  # Results
  cat("\nFully Bayesian Approach Results:\n")
  cat("-----------------------------------------------------------\n")
  cat("Number of Observations (k):", k, "\n")
  cat("Number of Trials per Observation (n):", n, "\n")
  cat("Prior Alpha (a):", alpha_prior, "\n")
  cat("Prior Beta (b):", beta_prior, "\n")
  cat("Posterior Alpha:", posterior_alpha, "\n")
  cat("Posterior Beta:", posterior_beta, "\n")
  cat("Computation Time:", computation_time, "seconds\n")
  cat("-----------------------------------------------------------\n\n")

  return(list(alpha = posterior_alpha, beta = posterior_beta, time = computation_time))
}

# Example
set.seed(42)
k <- 500
n <- 100
alpha_prior <- 2
beta_prior <- 3
theta <- rbeta(k, alpha_prior, beta_prior)
Y <- rbinom(k, n, theta)
bayesian_result <- fully_bayesian_posterior(Y, k, n, alpha_prior, beta_prior)
```

```
##
## Fully Bayesian Approach Results:
## -----------------------------------------------------------
## Number of Observations (k): 500
## Number of Trials per Observation (n): 100
## Prior Alpha (a): 2
## Prior Beta (b): 3
## Posterior Alpha: 1.989365
## Posterior Beta: 3.029162
## Computation Time: 2846.426 seconds
## -----------------------------------------------------------
```

```
polynomial_expansion_posterior <- function(Y, k, n, alpha_prior, beta_prior) {

  start_time <- Sys.time()

  S <- sum(Y)
  T <- k * n

  # Posterior moments approximation using polynomial expansion
  posterior_mean <- (alpha_prior + S) / (alpha_prior + beta_prior + T)
  posterior_variance <- ((alpha_prior + S) * (beta_prior + T - S)) /
                        (((alpha_prior + beta_prior + T)^2) * (alpha_prior + beta_prior + T + 1))
```

```r
  posterior_alpha <- posterior_mean * ((posterior_mean * (1 - posterior_mean)) / posterior_variance - 1)
  posterior_beta <- (1 - posterior_mean) * ((posterior_mean * (1 - posterior_mean)) / posterior_variance

  end_time <- Sys.time()
  computation_time <- difftime(end_time, start_time, units = "secs")

  #results
  cat("\nPolynomial Expansion Method Results:\n")
  cat("-------------------------------------------------------------\n")
  cat("Number of Observations (k):", k, "\n")
  cat("Number of Trials per Observation (n):", n, "\n")
  cat("Prior Alpha (a):", alpha_prior, "\n")
  cat("Prior Beta (b):", beta_prior, "\n")
  cat("Approximated Posterior Alpha:", posterior_alpha, "\n")
  cat("Approximated Posterior Beta:", posterior_beta, "\n")
  cat("Computation Time:", computation_time, "seconds\n")
  cat("-------------------------------------------------------------\n\n")

  return(list(alpha = posterior_alpha, beta = posterior_beta, time = computation_time))
}

# Example
set.seed(42)
k <- 500
n <- 100
alpha_prior <- 2
beta_prior <- 3
# Generate random theta values from Beta distribution
theta <- rbeta(k, alpha_prior, beta_prior)
Y <- rbinom(k, n, theta)
polynomial_result <- polynomial_expansion_posterior(Y, k, n, alpha_prior, beta_prior)
```

```
##
## Polynomial Expansion Method Results:
## -------------------------------------------------------------
## Number of Observations (k): 500
## Number of Trials per Observation (n): 100
## Prior Alpha (a): 2
## Prior Beta (b): 3
## Approximated Posterior Alpha: 1.971
## Approximated Posterior Beta: 3.0295
## Computation Time: 3.910065e-05 seconds
## -------------------------------------------------------------
```