MS2505: Bayesian Statistics
Course Project

December 26, 2024

Name
Samuel Jonsson

Email
sajs19@student.bth.se

Person Nr.
19990415-5596

Program
DVAMI19h

1 Analysis problem

The analysis aims to estimate the proportion of emails classified as spam in a given dataset. Using Bayesian methods, the focus is on deriving a reliable estimate of this proportion while accounting for uncertainty. The simplicity of the approach allows for clear insights into the spam classification problem, although it abstracts away complexities like email content or contextual nuances. The objective is to provide a robust probabilistic framework for understanding the distribution of spam within the dataset.

2 Data Description

The dataset selected for this project was obtained from Kaggle and consists of labeled email messages, where each email is classified as either "spam" or "ham". The dataset contains the following columns:

- **Category**: A binary label indicating whether the email is spam (1) or ham (0).
- **Message**: The text content of the email.

The first 10 rows of the dataset are shown in Table 1. For this analysis, the focus is on the Category column, which serves as the response variable for the Bayesian classification model.

Table 1
`mail_data.csv` dataset first 10 rows

Category	Message
ham	"Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat..."
ham	Ok lar... Joking wif u oni...
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question (std txt rate) T&C's apply 08452810075over18's
ham	U dun say so early ho... U c already then say...
ham	"Nah I don't think he goes to usf, he lives around here though"
spam	"FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, £1.50 to rcv"
ham	Even my brother is not like to speak with me. They treat me like aids patent.
ham	As per your request ´Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
...	...

Using a python script, the labels are then converted to numerical values, where spam is represented as 1 and ham as 0. After preprocessing the data, the dataset contains a total of

747	Spam
4825	Ham
5572	Total Emails

These counts are used to update the beta distribution's prior parameters and compute the posterior distribution.

Then, using a Python script, the labels were converted to 1 if it was "spam" and 0 if it was "ham", for easier analysis.

3 Model

The model chosen for this analysis is a binomial likelihood model with a beta prior. This approach is particularly suitable for binary classification problems where the response variable is dichotomous, such as classifying emails as either "spam" (1) or "ham" (0).

The binomial likelihood model represents the probability of observing a given number of successes (spam) in a fixed number of trials (emails). The parameter of interest, θ, represents the probability of an email being classified as spam. Using this model allows us to estimate θ while accounting for uncertainty.

To complement the likelihood, we use a $Beta(1, 1)$ prior, which is non-informative and reflects a state of prior ignorance about θ. This choice aligns with common practices in Bayesian analysis, particularly when no strong prior knowledge exists. By combining the prior with the observed data, we compute the posterior distribution of θ, providing a probabilistic estimate of the spam probability.

In addition, Markov Chain Monte Carlo (MCMC) methods are employed to validate the results obtained from the closed-form posterior distribution. MCMC sampling allows us to approximate the posterior distribution of θ when the analytic computation becomes infeasible.

4 Results

This section presents the outcomes of the Bayesian analysis performed using a Beta-binomial model to estimate the probability (θ) that an email is classified as spam. Results from Monte Carlo and MCMC sampling methods are summarized and validated, followed by visualizations to illustrate the posterior distributions and sampling behavior.

4.1 Posterior Analysis

Monte Carlo sampling, performed with 10,000 draws from the posterior distribution of θ, provides a detailed summary of the key statistics in Table 2. The posterior mean estimate for θ is approximately 13.42%, with a narrow 95% credible interval of $[0.125, 0.143]$. This suggests a high degree of confidence in the estimate and indicates that, on average, 13.42% of emails are classified as spam.

Table 2
Monte Carlo Posterior Analysis Data

Posterior Mean	0.1341994
Posterior SD	0.004576791
95% Credible Interval	$[0.1252803, 0.1432051]$

4.2 MCMC Validation

To validate the results, MCMC sampling was performed using the `MCMCmetrop1R` function. As shown in Table 3, the posterior mean is consistent at 13.44%, with a 95% credible interval of $[0.125, 0.144]$. These results further confirm the robustness of the estimates and align closely with the Monte Carlo findings.

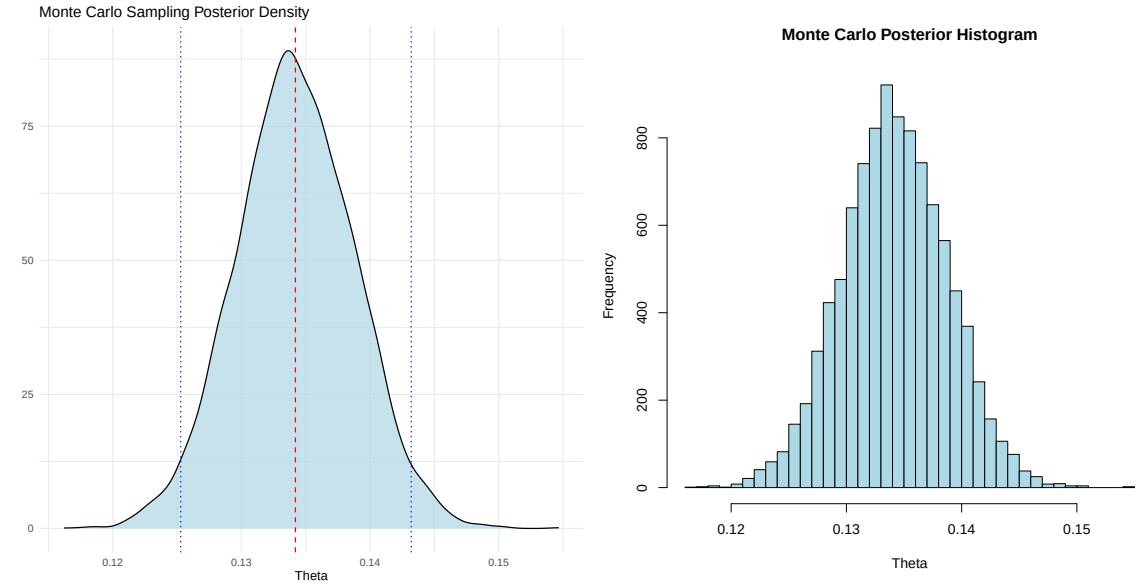
Table 3
MCMC Posterior Analysis Data

Posterior Mean	0.1344085
Posterior SD	0.004670508
95% Credible Interval	$[0.1254781, 0.1437142]$

The strong agreement between the Monte Carlo and MCMC methods demonstrates the model's validity and the consistency of the Bayesian inference process.

4.3 Visualization

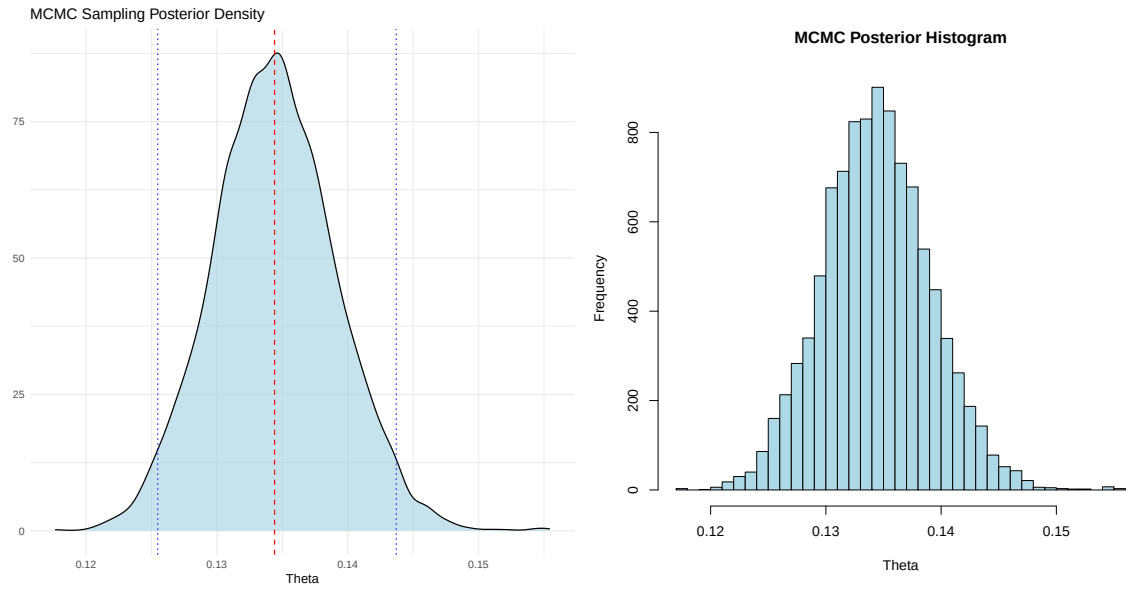
The posterior distributions derived from Monte Carlo and MCMC sampling are visualized in Figures 1 and 2, respectively. Both distributions exhibit a sharp peak around $\theta \approx 0.135$, reflecting the central tendency of the estimates.



(a) Density plot (b) Histogram

Figure 1
Posterior density and histogram for Monte Carlo sampling. The sharp peak at $\theta \approx 0.135$ aligns with Table 2.

Similarly, the MCMC posterior density and histogram (Figure 2) reveal consistent results, reinforcing the robustness of the Bayesian model.



(a) MCMC density plot

(b) MCMC histogram

Figure 2
Posterior density and histogram for MCMC sampling, demonstrating consistency with Monte Carlo results and Table 3.

The tails of the posterior distributions provide additional insights into the uncertainty associated with θ. Both Monte Carlo and MCMC density plots show symmetric tails that taper quickly beyond the 95% credible intervals, indicating a low likelihood of extreme values. Sparse population in the outer bins of the histograms further highlights the concentration of posterior mass near the central estimates. Slightly broader tails in the MCMC density plot compared to the Monte Carlo results may suggest minor differences in sampling variability, though these are negligible in the overall context of the model's consistency and reliability.

4.4 MCMC Trace Plot

The trace plot of the MCMC samples (Figure 3) provides further evidence of the sampling process's stability and convergence. The sampled values oscillate consistently around a central range without noticeable trends or significant deviations, confirming that the chain has reached a stationary distribution.

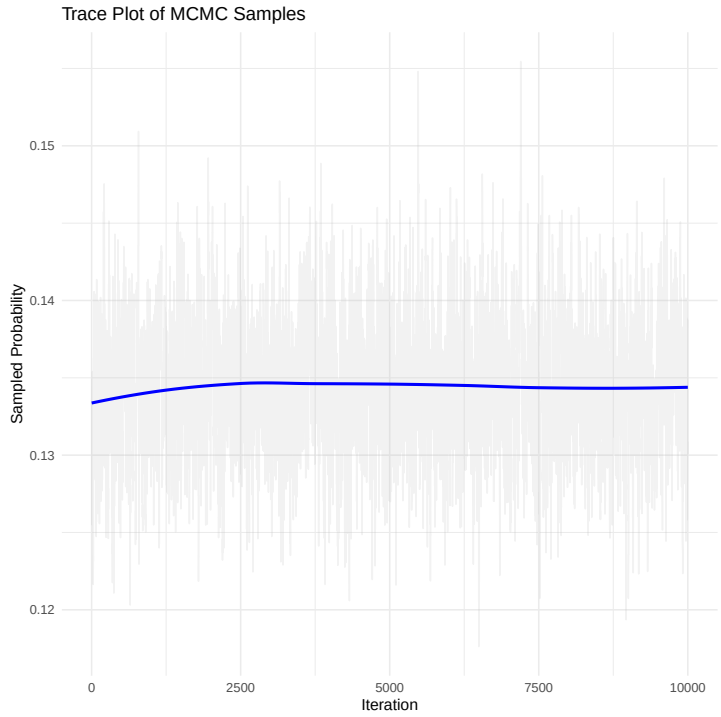


Figure 3
MCMC sampling trace plot. The stable fluctuations confirm convergence to the posterior distribution.

This stable behavior, along with minimal autocorrelation and effective mixing, suggests that the MCMC samples are representative of the posterior distribution. While minor clustering is observed in certain regions of the plot, this does not significantly affect the validity of the estimates.

5 Discussion

The Bayesian analysis conducted in this project produced reliable and consistent results for estimating the probability θ that an email is classified as spam. As mentioned in Section 3, the simplicity of the Beta-binomial model provided a clear and computationally efficient approach for this foundational analysis. Both Monte Carlo and MCMC sampling methods yielded posterior mean estimates of approximately 13.4%, with narrow 95% credible intervals, demonstrating a high level of confidence in the central estimate. The visualizations, discussed in Section 4.3, further supported these findings, revealing well-behaved posterior distributions with symmetric tails and rapid tapering, reflecting low uncertainty around the estimates. Additionally, the trace plot in Section 4.4 confirmed convergence of the MCMC sampling process, with sampled values stabilizing without significant deviations.

However, while the Beta-binomial model provides a straightforward and computationally efficient framework, it has inherent limitations. As mentioned in Section 3, the model focuses on binary classification, treating emails as either spam (1) or ham (0). This simplicity, while beneficial for interpretability, abstracts away complexities such as email content or metadata, which could provide richer insights into spam classification. Consequently, this approach restricts the model's ability to generalize to unseen datasets with varying characteristics. Additionally, the choice of a $Beta(1, 1)$ prior assumes no prior knowledge about the data, which, while this neutral stance aligns with the exploratory nature of the analysis, it may not optimally reflect datasets with unusual or heavily skewed properties.

To address these challenges, several improvements could be explored. First, as noted in Section 2, the dataset contains the email text content, and incorporating this into the model

could enhance its predictive capabilities, allowing it to better generalize to diverse or unknown datasets. For instance, incorporating natural language processing techniques to analyze the textual content could provide additional predictors beyond the binary classification framework. Additionally, extending the model to include hierarchical structures, as a refinement of the foundational framework described in Section 3, could capture variations across subpopulations or different email sources, offering a more nuanced understanding of θ.

The choice of prior distribution, discussed in Section 3, is another area for potential improvement. Rather than using a neutral $Beta(1, 1)$ prior, future analyses could explore data-informed priors that align with specific dataset characteristics. For instance, if prior information about spam prevalence is available from other sources, it could be incorporated to improve the robustness and sensitivity of the posterior analysis. The nature of the problem, however, would making a generally representative prior quite challenging, as the number of spam emails in an unknown dataset would be difficult to determine without more contextual knowledge.

On the computational side, the MCMC sampling methods, while effective, could benefit from the adoption of advanced techniques such as Hamiltonian Monte Carlo or adaptive MCMC methods. As noted in Section 4.2, MCMC sampling produced slightly broader tails compared to Monte Carlo sampling, suggesting some variability in the results. Enhanced sampling techniques could reduce this variability while improving computational efficiency.

In summary, while the simplicity of the beta-binomial model proved effective for this initial analysis, it highlights a trade-off between clarity and flexibility. The straightforward approach ensured interpretable results and computational feasibility. However, the challenges discussed here underscore the need for a more comprehensive framework in future analyses. Building on the current findings, incorporating richer features, hierarchical structures, and optimized priors would enhance the model's applicability to real-world spam detection scenarios. This iterative refinement process is essential for balancing simplicity with the complexities of dynamic and diverse datasets.

6 Appendix

A R Code

Listing 1
Project R code

```r
# ======================================================
# Bayesian Analysis for Email Spam Classification
# ======================================================

# Load Required Libraries
required_packages <- c("ggplot2", "dplyr", "MCMCpack", "coda", "
   tidyr")
installed_packages <- rownames(installed.packages())
for (pkg in required_packages) {
  if (!pkg %in% installed_packages) install.packages(pkg)
  library(pkg, character.only = TRUE)
}

# Set a global random seed for reproducibility
set.seed(123)

# Output Logs
if (!dir.exists("logs")) dir.create("logs")
results_log_file <- "logs/combined_results.log"

# ======================================================
# Load Data
# ======================================================

data_path <- "data/mail_data_bin.csv"
if (!file.exists(data_path)) stop("Data file not found!")
mail_data <- read.csv(data_path)

# Ensure the response variable is binary and properly coded
if (!all(mail_data$Category %in% c(0, 1))) {
  stop("The response variable 'Category' must be binary (0 or 1).")
}

# Metadata
spam_count <- sum(mail_data$Category == 1)
ham_count <- sum(mail_data$Category == 0)
total_emails <- nrow(mail_data)


sink(results_log_file)
cat("--- Data Metadata ---\n")
cat("Spam Count:", spam_count, "\n")
cat("Ham Count:", ham_count, "\n")
cat("Total Emails:", total_emails, "\n")
```

```
sink()

# ======================================================
# Beta Posterior Analysis
# ======================================================

# Prior parameters
prior_alpha <- 1
prior_beta <- 1

# Posterior parameters
posterior_alpha <- prior_alpha + spam_count
posterior_beta <- prior_beta + ham_count

# Monte Carlo Sampling
n_samples <- 10000
beta_samples <- rbeta(n_samples, posterior_alpha, posterior_beta)

# Summary statistics
beta_mean <- mean(beta_samples)
beta_sd <- sd(beta_samples)
beta_ci <- quantile(beta_samples, c(0.025, 0.975))

sink(results_log_file, append = TRUE)
cat("\n--- Beta Posterior Analysis ---\n")
cat("Posterior Mean:", beta_mean, "\n")
cat("Posterior SD:", beta_sd, "\n")
cat("95% Credible Interval:", beta_ci, "\n")
sink()

# ======================================================
# MCMC Sampling
# ======================================================

# Define log-posterior function
log_posterior <- function(params) {
  theta <- params[1]
  log_prior <- dbeta(theta, 1, 1, log = TRUE)
  log_likelihood <- sum(dbinom(mail_data$Category,
                               size = 1,
                               prob = theta,
                               log = TRUE))
  return(log_prior + log_likelihood)
}

# Run MCMC sampling
mcmc_results <- MCMCmetrop1R(
  fun = log_posterior,
  theta.init = 0.5,
  burnin = 1000,
  mcmc = n_samples,
```

```r
  thin = 1,
  verbose = 0
)

# Extract posterior samples
mcmc_samples <- as.vector(mcmc_results)
mcmc_mean <- mean(mcmc_samples)
mcmc_sd <- sd(mcmc_samples)
mcmc_ci <- quantile(mcmc_samples, c(0.025, 0.975))

sink(results_log_file, append = TRUE)
cat("\n--- MCMC Sampling Analysis ---\n")
cat("Posterior Mean:", mcmc_mean, "\n")
cat("Posterior SD:", mcmc_sd, "\n")
cat("95% Credible Interval:", mcmc_ci, "\n")
sink()

# ====================================================
# Diagnostics and Visualization
# ====================================================

if (!dir.exists("figures")) dir.create("figures")

# 1. Beta Density Plot
pdf("figures/beta_posterior_density_plot.pdf")
ggplot(data = data.frame(samples = beta_samples), aes(x = samples))
   +
  geom_density(fill = "lightblue", alpha = 0.7) +
  geom_vline(xintercept = beta_mean, color = "red", linetype = "
     dashed") +
  geom_vline(xintercept = beta_ci, color = "blue", linetype = "
     dotted") +
  labs(title = "Monte Carlo Sampling Posterior Density", x = "Theta
     ", y = "") +
  theme_minimal()
dev.off()

# 2. MCMC Density Plot
pdf("figures/mcmc_posterior_density_plot.pdf")
ggplot(data = data.frame(samples = mcmc_samples), aes(x = samples))
   +
  geom_density(fill = "lightblue", alpha = 0.7) +
  geom_vline(xintercept = mcmc_mean, color = "red", linetype = "
     dashed") +
  geom_vline(xintercept = mcmc_ci, color = "blue", linetype = "
     dotted") +
  labs(title = "MCMC Sampling Posterior Density", x = "Theta", y =
     "") +
  theme_minimal()
dev.off()

```

```r
# 3. Beta Posterior Histogram
pdf("figures/beta_posterior_histogram.pdf")
hist(beta_samples,
  breaks = 30, col = "lightblue", border = "black",
  xlab = "Theta", main = "Monte Carlo Posterior Histogram"
)
dev.off()

# 4. MCMC Posterior Histogram
pdf("figures/mcmc_posterior_histogram.pdf")
hist(mcmc_samples,
  breaks = 30, col = "lightblue", border = "black",
  xlab = "Theta", main = "MCMC Posterior Histogram"
)
dev.off()

# 5. Trace Plot for MCMC
pdf("figures/mcmc_trace_plot.pdf")
ggplot(
  data.frame(Iteration = 1:n_samples, Sample = mcmc_samples),
  aes(x = Iteration, y = Sample)
) +
  geom_line(alpha = 0.2, color = "gray") +
  geom_smooth(color = "blue", method = "loess", se = FALSE) +
  labs(title = "Trace Plot of MCMC Samples",
       x = "Iteration",
       y = "Sampled Probability") +
  theme_minimal()
dev.off()

# ======================================================
# Completion
# ======================================================

cat("Analysis complete. Check 'logs' and 'figures' directories for
   results.\n")
```