Bayesian Computations for Probit Models and Applications via Chinese Happyness Data

Travis Gubbe

December 17, 2025

Abstract

This report explores the applications of linear and nonparametric models with the addition of Bayesian estimation via Markov Chain Monte Carlo simulations to determine a binary response variable's status. Two spatial probit models - linear and varying coefficient - are constructed to classify a county in China as whether or not it is a high-ranking county based on socioeconomic and population factors. These models contained four explanatory variables: GDP, population per 10,000 residents, and two economic and financial factors. The models created demonstrate the role of spatial dependence in a variable's parameter estimation and the flexibility of regression splines to account for changes in time.

Introduction

The goal of the happyness data set presented is to analyze the various economic conditions and demographic factors leading to a county being ranked in the top 100 counties in China. The data set contains a variety of societal and economic explanatory variables, including GDP, population, savings balances of residents, and numerous fiscal and financial health ratios. The response variable is a binary variable, where if a county is in the top 100 counties, it is coded as a 1 - otherwise it is coded as a 0. The response variable is an unobservable latent variable, which is where the linear probit spatial model is key. A linear probit spatial model and varying coefficient spatial probit model are excellent options for an estimated model, which will be explained in the next section.

This reports aims to demonstrate the linear probit spatial model and varying coefficient spatial probit model to acquire Bayesian estimations and provide Bayesian quantities for the respective models. Bayesian estimations allow for flexibility in the modeling, particularly when accounting for spatial dependence. These models will demonstrate the power of spatial probit models and their relation to Bayesian statistics.

Probit Models

Linear Probit Spatial Model

Linear probit spatial models are useful when the response variable is binary and there is the presence of spatial dependence among the explanatory variables. Spatial dependence is described as a variable

in one location being influenced by a variable in another location, such as the GDP and population of a county being linked together. The model is constructed as follows:

$$y_{it}^* = \rho \sum_{j=1}^{n} \omega_{ij} y_{jt}^* + \beta_0 + \beta_1 X_{it} + \varepsilon_{it}, \qquad \varepsilon_{it} \sim N(0, \sigma_\varepsilon^2)$$

$$y_{it} = \begin{cases} 0, & y_{it}^* < 0, \\ 1, & y_{it}^* \geq 0. \end{cases}$$

ρ : spatial correlation parameter

ω : $N \times N$ spatial weight matrix

β : covariates used to approximate $m(x_{it})$

The spatial weights help define the strength of the units among one another, creating interdependence among the variables. The Markov Chain Monte Carlo Bayesian sampling will help smooth the latent variables and the complex likelihood function, making this a great potential model selection for the happyness data set.

Varying Coefficient Spatial Probit Model

Varying coefficient spatial probit model is similar to the linear spatial probit model, but captures change over time. Since the data set is from 2010-2020, this form of the model can be very useful in its design. The model is constructed as follows:

$$y_{it}^* = \rho \sum_{j=1}^{n} \omega_{ij} y_{jt}^* + \beta_0(t) + \beta_1(t) X_{it} + \varepsilon_{it}, \qquad \varepsilon_{it} \sim N(0, \sigma_\varepsilon^2).$$

where:

ρ : spatial correlation parameter

ω : $N \times N$ spatial weight matrix

β : spline basis functions

This allows the covariates to vary since the relationship among the predictors and outcomes may not be constant over time. Running Bayesian estimates on this model will smooth the covariates and spatial structure, making it a great option for the happyness data set. Unlike the linear version, the β values are spline functions to estimate the time-varying covariates, smoothing the model after many Markov Chain Monte Carlo simulations.

$$m(x_{it}) = \pi^(T) * (x_{it}) * \beta = \sum_{j=0}^{p} \phi_j x_{j,it} \; + \; \sum_{j=1}^{m} \phi_{j+p} \, (x_{it} - \xi_j)_+^p$$

This means running B-splines in conjunction with the covariates, with the assumption that the baseline of the propensity to be in the top 100 counties changes over time, but the effect of the covariates remains constant over time.

Case Study (Chinese Happyness Data)

Data Preparation

The Chinese happyness data set works well to demonstrate the two spatial probit models, since the response variable (top100) is a binary outcome with multiple variables that are spatially dependent with various socioeconomic variables. Before creating an appropriate model, the explanatory variables were checked for potential multicollinearity. Strong multicollinearity hints at dependency among the explanatory variables, making it difficult to discern the true effect of a variable on the result. Multicollinearity can be investigated by a correlation plot and a Variance Inflation Factor (VIF) plot. High correlations (> 0.8) hint at multicollinearity, while VIF values above 10 are strong indicators of multicollinearity. After running and re-running the variables through these two models, the chosen explanatory variables for modeling are - GDP, population per 10,000 residents, cy, and jr. These predictors are used in both the linear and varying coefficient spatial probit models.

Once the appropriate explanatory variables are selected, the variables are scaled to standardize the values and ensure the variables contribute equally to the model. This is particularly useful since there are high-value variables (GDP, population) and low-value variables (cy, jr), which can cause the low-value variables to be undervalued in the final model interpretation if they are not scaled. In addition, scaling the data can help the model run faster and improve model performance for distance-related models, as is the case with the spatial probit models.

For illustrative purposes of the comparison of the linear probit spatial model and the varying coefficient spatial probit model will be run with the same explanatory variables. This is to demonstrate both model's effectiveness when using the same parameters and their performance to one another for this data set.

Results

First, a linear spatial probit model is created to predict if a county is in the top 100 counties with the four explanatory variables listed (GDP, population, cy, jr):

Variable	**Estimate**	**Std. Dev**	**95% CI**
Intercept	-0.261	0.067	[-0.393, -0.128]
GDP	0.319	0.088	[0.148, 0.491]
Population (per 10K)	0.432	0.087	[0.262, 0.601]
cy	0.357	0.085	[0.189, 0.524]
jr	0.198	0.076	[0.049, 0.346]
ρ	0.619	0.0402	[0.540, 0.698]

Table 1: Summary table for the Linear Spatial Probit Model. All variables are statistically signficant since each respective confidence interval does not contain 0.

From the results, GDP, population, cy, and jr all have positive estimates and their credible intervals do not contain zero, which means these factors have a positive impact on whether a county will be ranked in the top 100 and are statistically significant to the model. The estimates also provide the statistical impact of each individual variable - for example, GDP's estimate of 0.319 means, assuming all other variables are fixed, increasing the GDP by one unit increases the probability of the response variable (top 100) by 0.319. The ρ value's estimate value of 0.619 and 95% credible

interval of $[0.540, 0.698]$ indicates there is a positive strong spatial dependence in the model, meaning county's locations to one another can impact their determination of whether they are a top 100 county. This means that a county ranked in the top 100 is more likely to be surrounded by other high-ranking counties. From the model results, these highly-ranked counties may be clustered together geographically, with their ranking being explained by GDP, population, and cy and jr factors.

Since the variables present can change over a period of time, the varying coefficient spatial probit model is utilized to account for the change in time with the use of regression splines. It was assumed the baseline of the propensity to be ranked in the top 100 counties in China would vary over time, while the covariate factors would remain relatively constant for the 10-year time span. This adds only the regression spline values to the model and shortens the risk of overfitting, while keeping the flexibility of the model and makes for easier interpretability. Multiple models were run with varying spline amounts and degrees between 1 and 4. In the end, it was determined to use 1 regression spline in the model with 1 degree to retain statistical significance of the four chosen variables at the 95% significance level. In addition, model performance testing showed very little change in the model performance metrics to confidently change the amount of regression splines and degrees of the model.

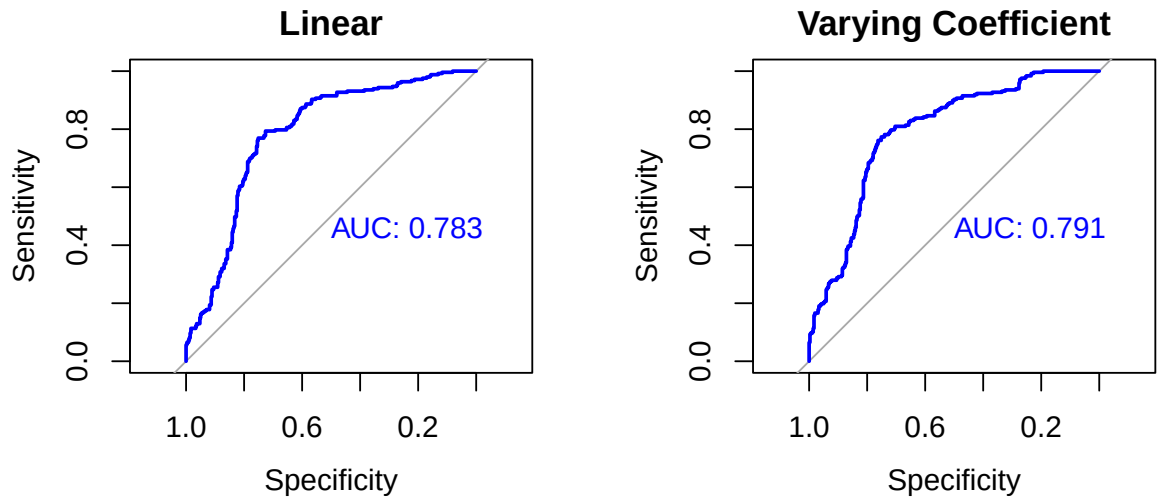
From the results table for the varying coefficient model, the covariate estimates differ slightly to the linear model, while the ρ value is equal to the linear model, with both values equal to 0.619. All covariates are statistically significant at the 95% significance level, which can also be confirmed by the credible intervals for each covariate as none contain zero **(Appendix B)**. Now that two appropriate spatial probit models have been created, there are a couple of model performance diagnostics that can be run to determine which model performed best for categorizing a top 100 county in China.

Trace plots are also viewed to determine as a diagnostic tool for the spatial probit models. A good trace plot has a fuzzy look, with no clear pattern in the plot. This would indicate good convergence of the MCMC simulations for both the regression and spatial parameters. The trace plots for each spatial model showed random traces and no clear pattern, which indicates good convergence of the MCMC simulations for both models **(Appendix C)**.

The density plots for the posterior distributions of the covariates of both models can also be viewed as a diagnostic approach. The goal is to have unimodal density plots and a reasonable spread for each plot (preferably non-zero), indicating each covariate has an association with the response variable and is easy to interpret. The density plots for both the linear spatial probit model and varying coefficient were mostly unimodal in appearance with a good spread in the density as well as what appear to be non-zero covariates, indicating the covariates selected are statistically significant and have an identifiable association with the response variable **(Appendix C)**.

Model Performance

Now that the two models have been created, they must be assessed to determine their performance to one another. The two methods in this paper for model comparison is Deviance Information Criterion (DIC) and a combination of Receiver Operating Characteristic (ROC) with Area Under the Curve (AUC). DIC is a generalization of AIC and is used to compare Bayesian models. The goal is to have a lower DIC value, as this is an indication of model fit and complexity. In contrast, it is best to have a model with higher ROC and AUC scores, as this indicates the model can explain a higher amount of variability and indicates better model performance.



Linear DIC: 692.2 **VC DIC:** 681.51

Figure 1: ROC visualizations for the two models with their respective AUC values. The Varying Coefficient model performed slightly better, with an AUC value of 0.791 compared to the Linear Spatial model with an AUC value of 0.783. The DIC values show the Varying Coefficient model performed better than the Linear Spatial model, with a lower DIC value of 680.78 compared to 691.08

From the ROC/AUC results, the Varying Coefficient model had a slightly higher AUC value of 0.791 compared to the Linear spatial model's AUC value of 0.783. An AUC value of close to 0.8 indicates very good model performance, which means both models have performed very well in the prediction of whether a county in China is in the top 100. In addition, with a DIC value of 680.78 compared to a DIC value of 691.08, the Varying Coefficient model performed better than the Linear spatial model in regards to DIC, as the lower DIC value (especially since the DIC difference is greater than 10) indicates better model fit and performance. It is important to note the standard threshold value of 0.5 was chosen for the ROC. Optimization of the ROC threshold can lead to a better AUC as well as a better performance model as it pertains to Type I and Type II error. Doing so can also change the outlook of which model type fits better for the problem at hand, as optimizing the ROC threshold impacts both models' performance. For the purpose of this paper, ROC optimization was not explored since the objective is to provide working probit spatial models.

Conclusion

The results of the linear probit spatial model and varying coefficient spatial probit in conjunction with Bayesian estimation highlight its role in determining spatial dependencies for regional outcomes. The estimated spatial correlation parameter (ρ) poses the observed outcomes in county ranking positively influences its neighboring regions, implying strong spatial interaction. This is an important distinction made by the spatial probit models, as its indicates a county's socio-economic factors can influence neighboring regions. From the two models created, the varying coefficient spatial probit model displayed stronger model performance than its counterpart, as the β-splines

obtain non-parametric relationships between the covariates and the outcome, which is particularly essential for relationships that are not linear.

Both models had a ρ value equal to 0.619, which indicates a strong positive correlation between neighboring counties in its determination of being ranked in the top 100 counties, demonstrating strong spatial dependency. In other words, counties in the top 100 tend to be regionally close together rather than far apart. Both models also have positive estimate parameters for all selected covariates, meaning a 1 unit increase in each covariate has a positive impact on the outcome. In addition, all covariates' credible intervals do not contain zero, which indicates the covariates are statistically significant for both models. Where the varying coefficient model separated itself from the linear model is its AUC and DIC values. The varying coefficient model had a higher AUC value (0.791 vs. 0.783) and a lower DIC value (680.78 vs. 691.08) which indicates the varying coefficient model captured more variability and better balance when fitting the data.

In the end, both models demonstrate the ability to provide practical insights into spatial dependency for binary regional outcomes. The linear spatial probit model is easier to construct and interpret, while the non-parametric spatial probit model incorporates regression splines to estimate time-varying coefficients. For an alternative model, interaction effects can potentially create a more accurate model if it is believed the impact of one covariate affects the value of another covariate. Overall, probit models are a powerful tool in spatial modeling, and can provide applicable insights in regional outcomes.

Appendix

Appendix A

Variable	VIF
GDP	1.48
Population (per 10K)	1.10
cy	1.29
jr	1.16

Table 2

Appendix B

Variable	Estimate	Std. Dev	95% CI
Intercept	0.149	0.168	[-0.180, 0.478]
GDP	0.472	0.124	[0.230, 0.714]
Population (per 10K)	0.461	0.091	[0.283, 0.638]
cy	0.201	0.099	[0.006, 0.396]
jr	0.280	0.090	[0.103, 0.457]
B1	-0.794	0.325	[-1.430, -0.158]
ρ	0.619	0.041	[0.540, 0.698]

Table 3: Summary table for the Varying Coefficient Spatial Probit Model. All covariates are statistically significant at the 95% confidence level since they do not contain 0.

Appendix C

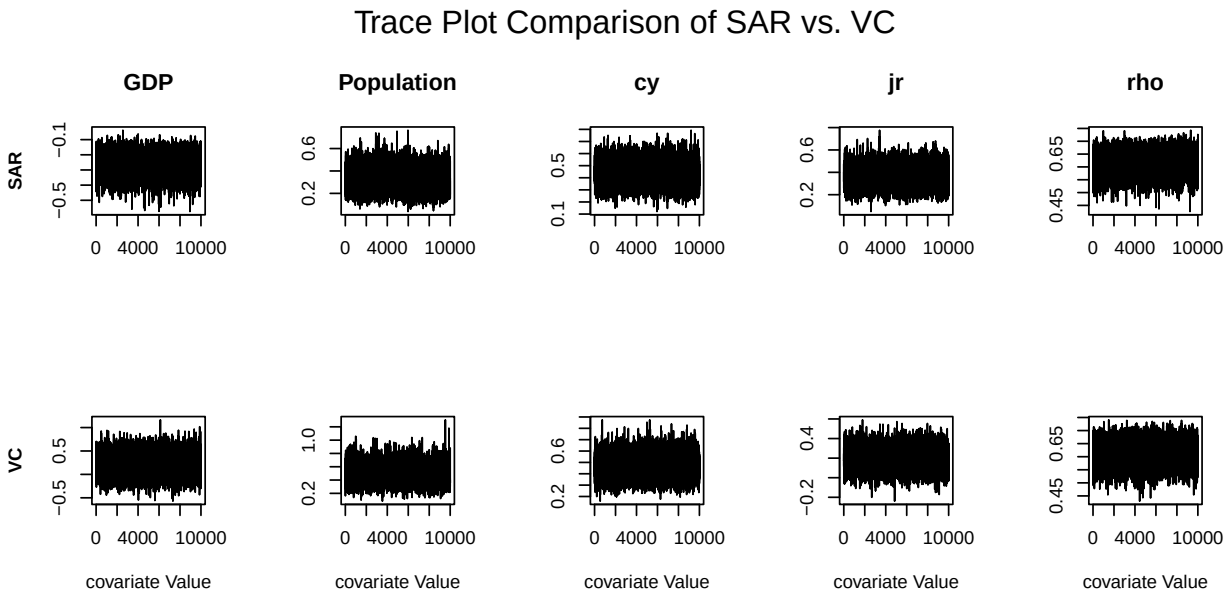


Figure 2: Trace plots of the covariates for the linear spatial probit model and varying coefficient probit spatial model, illustrating good convergence of MCMC simulation for both models.

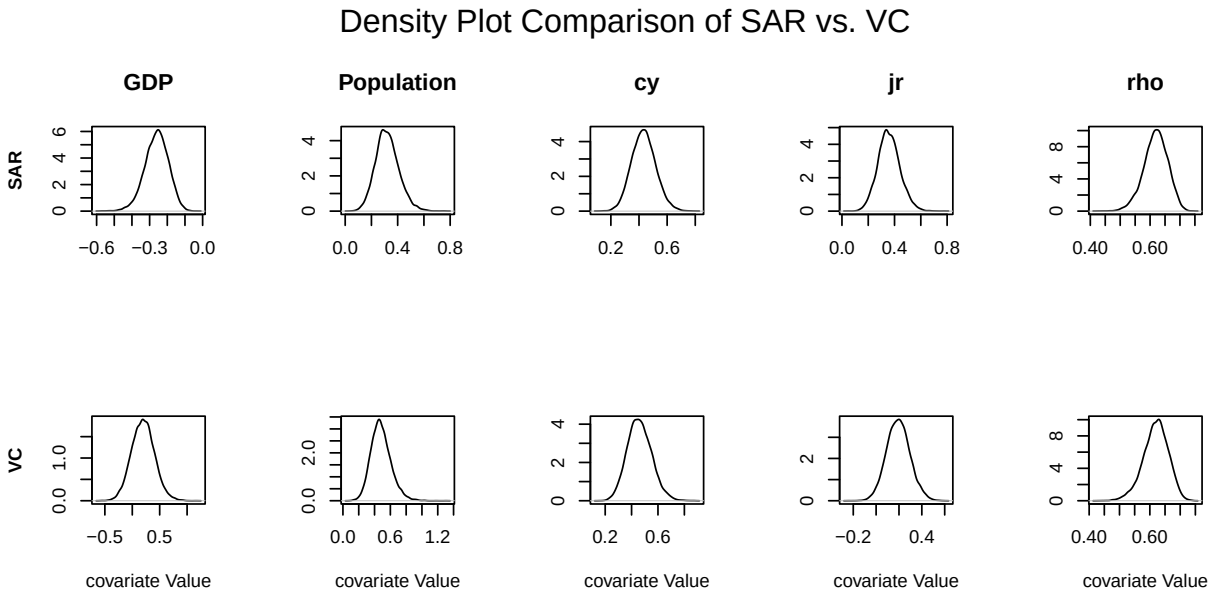


Figure 3: Density plots of the covariates for the linear spatial probit model and varying coefficient probit spatial model, illustrating good convergence of MCMC simulation for both models.

R Code

```r
library(spatialprobit)
library(Matrix)
library(dplyr)
library(splines)

happy <- read.csv("happying_english.csv", stringsAsFactors = FALSE)

# Y variable is numeric
happy$top100 <- as.numeric(happy$top100)

# Create IDs for counties and years, then sort (year, county)
counties <- sort(unique(happy$county_name))
years    <- sort(unique(happy$year))

happy <- happy %>%
  mutate(
    county_id = match(county_name, counties),
    year_id   = match(year, years)
  ) %>%
  arrange(year_id, county_id)

N  <- length(counties)   # number of counties
TT <- length(years)      # number of years
stopifnot(N * TT == nrow(happy))

# Cross-sectional W_N (N x N) sparse
W_N <- Matrix(0, nrow = N, ncol = N, sparse = TRUE)

for (i in 1:N) {
  city_i <- happy$city[happy$county_id == i][1]
  for (j in 1:N) {
    if (i != j) {
      city_j <- happy$city[happy$county_id == j][1]
      if (!is.na(city_i) && !is.na(city_j) && city_i == city_j) {
        W_N[i, j] <- 1
      }
    }
  }
}

# Zero diagonal
diag(W_N) <- 0

# Row-standardize: rows sum to 1 when neighbors exist
rs <- rowSums(W_N)
rs[rs == 0] <- 1
W_N <- Diagonal(x = 1 / rs) %*% W_N

# Panel W = I_T x W_N (block-diagonal, sparse)
I_T     <- Diagonal(TT)
W_panel <- kronecker(I_T, W_N)

selected_vars <- c("GDP", "registered_pop_10k", "cy", "jr")

```

```r
Xvars <- selected_vars
# 1) Extract predictors as numeric matrix
X_no_int <- as.matrix(happy[, Xvars])

# Make sure everything is numeric
X_no_int <- apply(X_no_int, 2, as.numeric)

# 2) SCALE the predictors: center = TRUE, scale = TRUE
X_scaled <- scale(X_no_int, center = TRUE, scale = TRUE)

# 3) Add intercept AFTER scaling
X <- cbind(Intercept = 1, X_scaled)
colnames(X)[1] <- "Intercept"

# Response
y <- happy$top100

set.seed(123)

# Linear Spatial Probit Model
############################################
fit_sar <- sar_probit_mcmc(
  y      = y,
  X      = X,
  W      = W_panel,
  ndraw  = 10000,
  burn.in = 2000,
  thinning = 5,
  prior  = NULL,
  showProgress = TRUE
)

# 'fit_sar' is an object of class 'sarprobit'
summary(fit_sar)
# diagnostic plots: chains, densities, ACF, etc.
plot(fit_sar)

mean(fit_sar$rho) #p rho = 0.619

rho_mean <- 6.187e-01
rho_sd   <- 4.024e-02

rho_ci_norm <- rho_mean + qnorm(c(0.025, 0.975)) * rho_sd
rho_ci_norm #[0.540, 0.698]

intercept_ci_norm <- -0.26051 + qnorm(c(0.025, 0.975)) * 0.06785
gdp_ci_norm <- 0.31924 + qnorm(c(0.025, 0.975)) * 0.08762
pop_ci_norm <- 0.43169 + qnorm(c(0.025, 0.975)) * 0.08662
cy_ci_norm <- 0.35673 + qnorm(c(0.025, 0.975)) * 0.08540
jr_ci_norm <- 0.19793 + qnorm(c(0.025, 0.975)) * 0.07576

intercept_ci_norm #(-0.393, -0.128)
gdp_ci_norm #(0.148, 0.491)
pop_ci_norm #(0.262, 0.601)
cy_ci_norm #(0.189, 0.524)
jr_ci_norm #(0.049, 0.346)
```

```r

#Varying Coefficient Spatial Probit Model
#####################################################
# Response is numeric
happy$year <- as.numeric(happy$year)

# Scale time to [0, 1]
t_min <- min(happy$year)
t_max <- max(happy$year)

happy$t_scaled <- (happy$year - t_min) / (t_max - t_min)

# Number of splines (B1)
K <- 1
#1 Degree
B <- bs(happy$t_scaled, df = K, degree = 1)
B <- as.matrix(B)
colnames(B) <- paste0("B", 1:K)

# Attach to data frame
for (k in 1:K) {
  happy[[paste0("B", k)]] <- B[, k]
}

spline_terms <- paste0("B", 1:K)

vars_static <- c("GDP", "registered_pop_10k", "cy", "jr")

# Extract and coerce to numeric
X_static_raw <- happy[, vars_static, drop = FALSE]
X_static_raw[] <- lapply(X_static_raw, as.numeric)

# Scale for numerical stability
X_static_scaled <- scale(as.matrix(X_static_raw))
colnames(X_static_scaled) <- vars_static

# GDP as numeric
#happy$GDP <- as.numeric(happy$GDP)

# Scale GDP
X_vc <- cbind(
  Intercept = 1,
  X_static_scaled,                    # constant effects
  as.matrix(happy[, spline_terms])    # varying intercept basis
)

X_vc <- as.matrix(X_vc)
storage.mode(X_vc) <- "double"

fit_vc <- sar_probit_mcmc(
  y      = y,
  X      = X_vc,
  W      = W_panel,
  ndraw  = 10000,
  burn.in = 2000,
```

```
  thinning = 5,
  prior  = NULL,        # or your prior list
  showProgress = TRUE
)

summary(fit_vc)

plot(fit_vc)

fit_vc$bdraw
#p rho = 0.620
rho_mean_vc <- 6.19e-01
rho_sd_vc   <- 4.0128e-02

rho_ci_norm_vc <- rho_mean_vc + qnorm(c(0.025, 0.975)) * rho_sd_vc
rho_ci_norm_vc #[0.540, 0.698]

int_ci_norm_vc <- 0.14883 + qnorm(c(0.025, 0.975)) * 0.16784
int_ci_norm_vc #[-0.180, 0.478]

gdp_ci_norm_vc <- 0.47206 + qnorm(c(0.025, 0.975)) * 0.12366
gdp_ci_norm_vc #[0.230, 0.714]

pop_ci_norm_vc <- 0.46081 + qnorm(c(0.025, 0.975)) * 0.09051
pop_ci_norm_vc #[0.283, 0.638]

cy_ci_norm_vc <- 0.20094 + qnorm(c(0.025, 0.975)) * 0.09928
cy_ci_norm_vc #[0.006, 0.396]

jr_ci_norm_vc <- 0.28026 + qnorm(c(0.025, 0.975)) * 0.09023
jr_ci_norm_vc #[0.103, 0.457]

b1_ci_norm_vc <- -0.79404 + qnorm(c(0.025, 0.975)) * 0.32471
b1_ci_norm_vc #[-1.430, -0.158]

####################
library(pROC)

predict_prob_probit_approx <- function(fit, X) {
  beta_hat <- colMeans(fit$bdraw)
  eta_hat  <- as.vector(X %*% beta_hat)
  p_hat    <- pnorm(eta_hat)
  pmin(pmax(p_hat, 1e-12), 1 - 1e-12)
}

p_lin <- predict_prob_probit_approx(fit_sar, X)
roc_lin <- roc(response = y, predictor = p_lin)
auc_lin <- auc(roc_lin) #0.7833
plot(roc_lin, plot = TRUE, col = "blue", print.auc = TRUE, main = "Linear")

p_vc <- predict_prob_probit_approx(fit_vc, X_vc)
roc_vc <- roc(response = y, predictor = p_vc)
auc_vc <- auc(roc_vc) #0.791

plot(roc_vc, plot = TRUE, col = "blue", print.auc = TRUE, main = "Varying
    Coefficient")
```

```r
#Follow-up
################################################
compute_dic_probit_approx <- function(fit, X, y, ndraw_eval = 1000) {
  # X: design matrix used in fitting (with intercept)
  # y: 0/1 outcome vector

  nd <- min(ndraw_eval, nrow(fit$bdraw))
  idx <- seq(1, nrow(fit$bdraw), length.out = nd)

  dev_vec <- numeric(nd)

  for (k in seq_along(idx)) {
    j <- idx[k]
    beta_j <- fit$bdraw[j, ]

    eta_j <- as.vector(X %*% beta_j)
    p_j   <- pnorm(eta_j)
    p_j   <- pmin(pmax(p_j, 1e-12), 1 - 1e-12)

    loglik_j <- sum(y * log(p_j) + (1 - y) * log(1 - p_j))
    dev_vec[k] <- -2 * loglik_j
  }

  # Posterior mean deviance
  D_bar <- mean(dev_vec)

  # Deviance at posterior mean beta
  beta_hat <- colMeans(fit$bdraw)
  eta_hat  <- as.vector(X %*% beta_hat)
  p_hat    <- pnorm(eta_hat)
  p_hat    <- pmin(pmax(p_hat, 1e-12), 1 - 1e-12)

  loglik_hat <- sum(y * log(p_hat) + (1 - y) * log(1 - p_hat))
  D_hat <- -2 * loglik_hat

  # DIC
  DIC <- 2 * D_bar - D_hat

  list(D_bar = D_bar, D_hat = D_hat, DIC = DIC)
}

DIC_lin <- compute_dic_probit_approx(fit_sar, X, y)
DIC_lin$DIC #691.08

DIC_vc <- compute_dic_probit_approx(fit_vc, X_vc, y)
DIC_vc$DIC #680.78

#Plots
################################################################
#Trace plots for SAR and VC
par(mfrow=c(2,5), oma=c(0,0,2,0))
plot(fit_sar$bdraw[,1], type = "l",
     main = "GDP",
     xlab = "",
     ylab = expression(bold("SAR")))
plot(fit_sar$bdraw[,2], type = "l",
```

```
     main = "Population",
     xlab = "",
     ylab = "")
plot(fit_sar$bdraw[,3], type = "l",
     main = "cy",
     xlab = "",
     ylab = "")
plot(fit_sar$bdraw[,4], type = "l",
     main = "jr",
     xlab = "",
     ylab = "")
plot(fit_sar$pdraw, type = "l",
     main = "rho",
     xlab = "",
     ylab = "")
plot(fit_vc$bdraw[,1], type = "l",
     main = "",
     xlab = "covariate Value",
     ylab = expression(bold("VC")))
plot(fit_vc$bdraw[,2], type = "l",
     main = "",
     xlab = "covariate Value",
     ylab = "")
plot(fit_vc$bdraw[,3], type = "l",
     main = "",
     xlab = "covariate Value",
     ylab = "")
plot(fit_vc$bdraw[,4], type = "l",
     main = "",
     xlab = "covariate Value",
     ylab = "")
plot(fit_vc$pdraw, type = "l",
     main = "",
     xlab = "covariate Value",
     ylab = "")
mtext("Trace Plot Comparison of SAR vs. VC",
      outer = TRUE, cex = 1.2)
#Density plots for SAR and VC
par(mfrow=c(2,5), oma=c(0,0,2,0))
plot(density(fit_sar$bdraw[,1]),
     main = "GDP",
     xlab = "",
     ylab = expression(bold("SAR")))
plot(density(fit_sar$bdraw[,2]),
     main = "Population",
     xlab = "",
     ylab = "")
plot(density(fit_sar$bdraw[,3]),
     main = "cy",
     xlab = "",
     ylab = "")
plot(density(fit_sar$bdraw[,4]),
     main = "jr",
     xlab = "",
     ylab = "")
plot(density(fit_sar$pdraw),
```

```
     main = "rho",
     xlab = "",
     ylab = "")
plot(density(fit_vc$bdraw[,1]),
     main = "",
     xlab = "covariate Value",
     ylab = expression(bold("VC")))
plot(density(fit_vc$bdraw[,2]),
     main = "",
     xlab = "covariate Value",
     ylab = "")
plot(density(fit_vc$bdraw[,3]),
     main = "",
     xlab = "covariate Value",
     ylab = "")
plot(density(fit_vc$bdraw[,4]),
     main = "",
     xlab = "covariate Value",
     ylab = "")
plot(density(fit_vc$pdraw),
     main = "",
     xlab = "covariate Value",
     ylab = "")
mtext("Density Plot Comparison of SAR vs. VC",
      outer = TRUE, cex = 1.2)
#Diagnostics
############################################################
X <- happy[, Xvars]
X[] <- lapply(X, as.numeric)

drop_vars <- c("top100", "county_name", "city", "province", "county_id", "year_id")
drop_vars <- intersect(drop_vars, names(happy))

Xvars     <- setdiff(names(happy), drop_vars)

Xvars # check predictors

round(cor(X_scaled), 2)
# correlation matrix
cor_X <- cor(X, use = "pairwise.complete.obs")
round(cor_X, 2)

library(car)

fm2 <- lm(
  as.numeric(top100) ~ .,
  data = cbind(top100 = happy$top100, happy[, selected_vars])
)

vif_values2 <- vif(fm2)

sort(vif_values2, decreasing = TRUE)

#Scaling the selected variables and checking for multicollinearity
sel_vars <- happy[, selected_vars]              # your chosen predictors
sel_vars[] <- lapply(sel_vars, as.numeric)     # make sure numeric
```

```r
sel_vars_scaled <- scale(sel_vars)

sel_vars_scaled_df <- as.data.frame(sel_vars_scaled)

lm_scaled_2 <- lm(
  as.numeric(top100) ~ .,
  data = cbind(top100 = happy$top100, sel_vars_scaled_df)
)

vif_vals2 <- vif(lm_scaled_2)
vif_vals2
```

Listing 1: R Source Code