BAYESIAN DATA ANALYSIS

MINI-PROJECT

Members :

PHOEBE PATRICIA WIBOWO - 2602080825

JENNIFER ARDELIA LIMICIA - 2602105090

CRYSANTHA MONICA LIM - 2602090076

RANEL DAIYANJABBAR - 2602088443

GLADYS LIONARDI - 2602073076

CHERYLENE CALLISTA REKSOHARTONO - 2602087024

I. Introduction

Dataset "Wine Quality" yang kami miliki berasal dari Kaggle (yang diunduh dari UCI Machine Learning Repository oleh pengunggah). Dataset ini berisi informasi tentang varian merah dan putih dari anggur Portugis "Vinho Verde". Terdapat 13 variabel dalam dataset ini, di mana 11 di antaranya (*fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol*) adalah variabel fisikokimia (input), sedangkan dua variabel lainnya adalah sensorik (*quality* & *type*). Semua variabel memiliki nilai numerik, kecuali jenis anggur.

II. Models

Model Bayesian ini difokuskan untuk menentukan variabel yang berpengaruh terhadap kualitas anggur, dengan menggunakan kovariat yang mencakup fitur-fitur fisiko-kimia tertentu, yaitu fixed.acidity, volatile.acidity, citric.acid, residual.sugar, chlorides, pH, dan sulphates.

Untuk model pertama, distribusi prior yang digunakan yaitu Normal dengan rata - rata 0 dan varians 0.01. Lalu untuk likelihood menggunakan distribusi Bernoulli. Setelah itu, kami menjalankan model kedua (menggunakan seleksi fitur dari SSVS) dengan prior yang menggunakan distribusi Bernoulli dengan probabilitas 0.5 (untuk parameter γ) dan Normal (untuk parameter δ) dengan rata - rata 0 dan varians 0.01 untuk memilih variabel mana saja yang mempengaruhi kualitas dari wine secara signifikan. Kemudian model ketiga menggunakan distribusi Normal dengan rata - rata 0 dan varians 1 sebagai prior dan distribusi Bernoulli sebagai likelihood dengan fungsi logit. Ketiga model merupakan model logistic regression dengan output binary (0 dan 1). Logistic regression model yang digunakan adalah

$$\text{logit}(q_i) = \eta_i = \sum_{j=1}^{J} X_{ij}\beta_j$$

Input Data:

- Y adalah variabel respons, bernilai 1 atau 0, dan merupakan kualitas anggur.
- X adalah matriks kovariat yang mencakup fitur-fitur fisiko-kimia yang diinginkan.
- n adalah jumlah observasi dalam dataset.

Model ini dirancang untuk memahami pengaruh variabel fisiko-kimia (yang sudah disebutkan di atas) pada kualitas anggur menggunakan pendekatan Bayesian, dengan mengasumsikan bahwa distribusi prior dari parameter bersifat normal dan distribusi likelihood bersifat Bernoulli.

III. Computation

Dalam analisis Bayesian menggunakan *software* JAGS melalui paket rjags pada R, beberapa *key parameter* harus diperhatikan untuk memperoleh hasil yang optimal. Proses ini melibatkan tahap *burn-in*, *post-burn in sample*, jumlah rantai Markov Chain Monte Carlo (MCMC), dan interval pengambilan sampel *(thinning)*. Sebanyak 1000 iterasi awal digunakan

sebagai tahap *burn* lalu dilakukan 5000 iterasi tambahan untuk menghasilkan sampel yang merepresentasikan distribusi posterior dari parameter model, dengan 2 chain dan thinning interval sebesar 5.

Konvergensi diperiksa untuk ketiga model, menggunakan Geweke (|value| < 2 menunjukkan konvergensi) dan Gelman-Rubin (1 menunjukkan konvergensi). Dengan Geweke, pada model pertama sebagian besar variabel bernilai mutlak lebih kecil daripada 2 (ada 1 variabel, yakni pH, yang bernilai lebih dari 2), yang berarti sebagian besar sudah bersifat konvergen. Pada model kedua dan ketiga, nilai mutlak seluruh beta tidak ada yang lebih dari 2. Jadi dalam sisi konvergensi berdasarkan Geweke, model 1 tidak sebaik 2 model lainnya.

Hasil dari Gelman-Rubin untuk model pertama dan kedua memiliki hasil yang baik, dimana keduanya memiliki parameter yang bernilai sekitar 1-1.02. Untuk model ketiga, nilai-nilainya sedikit lebih buruk dibanding kedua model pertama, namun masih dibawah 1.1 sehingga masih bisa dikatakan konvergen.

Kami kemudian menggunakan autocorrelation untuk memeriksa apakah per variabelnya bersifat independen dan tidak saling berkorelasi. Hasil autokorelasi pada kedua chain dalam masing-masing model sangatlah kecil, dengan nilai autokorelasi mutlak terbesar 0.09 di model pertama , 0.04 di model kedua (ini model dengan nilai autokorelasi terkecil) dan sedikit di atas 0.1 untuk model ketiga, sehingga dapat dibuktikan bahwa tidak ada ketergantungan antar variabel.

Berikut ini adalah hasil aproksimasi beta untuk ketiga model. Karena model kedua hanya menggunakan variabel tertentu, tidak terdapat beta untuk semua variabel.

Tabel 1 Estimasi kovariat beta dari ketiga model

	Model 1	Model 2	Model 3
Y Average	0.6331	0.6330	0.6329
β fixed.acidity	0.1028		0.0971
β volatile.acidity	-0.7311	-0.7132	-0.7308
β citric.acidity	-0.0837		-0.0834
β residual.sugar	0.4176	0.3011	0.4101
β chlorides	-0.0332		-0.0336
β free.sulfur.dioxide	0.2980	0.2992	0.2927
β total.sulfur.dioxide	-0.4379	-0.4329	-0.4317
β density	-0.1734		-0.1625
β pH	0.1128		0.1083
β sulphates	0.3167	0.2801	0.3154
β alcohol	1.0399	1.1361	1.0437

IV. Model comparisons

Model pertama dan ketiga menggunakan 11 variabel input (mengecualikan tipe wine karena tidak relevan) tanpa terkecuali. Berdasarkan hasil SSVS, khususnya kolom INC

probability, kami melakukan seleksi ulang terhadap variabel - variabel yang ada. Seperti yang dapat dilihat, kovariat variabel volatile acidity, residual sugar, free sulfur dioxide, total sulfur dioxide, sulphates dan alcohol bernilai 1 yang berarti variabel tersebut memiliki pengaruh yang signifikan dalam memprediksi variabel respons yaitu kualitas wine. Maka dari itu, kami menjalankan model kedua hanya dengan 6 variabel tersebut.

Tabel 2 Summary posterior dari SSVS

	INC	**50%**	**5%**	**95%**
β fixed.acidity	0.0039	0	0	0
β volatile.acidity	**1**	-0.7215	-0.7963	-0.6559
β citric.acidity	0.2122	0	-0.1232	0
β residual.sugar	**1**	0.3048	0.2446	0.3653
β chlorides	0.0228	0	0	0
β free.sulfur.dioxide	**1**	0.2986	0.2267	0.3708
β total.sulfur.dioxide	**1**	-0.4318	-0.5092	-0.3573
β density	0.0147	0	0	0
β pH	0.0674	0	0	0.0624
β sulphates	**1**	0.2834	0.2275	0.3442
β alcohol	**1**	1.1364	1.0673	1.2039

Performa dari ketiga model dibandingkan dengan Deviance Information Criteria (DIC), Watanabe-Akaike Information Criteria (WAIC), serta posterior-predictive p-value (PPP). Tabel berikut merupakan tabel perbandingan ketiga model tersebut.

Tabel 3 Perbandingan performa model

	Model 1	Model 2	Model 3
DIC	6728	6735	6728
WAIC	6729.875	6735.902	6729.285
PPP	0.4896	0.4914	0.4936

Dapat dilihat pada tabel di atas, model 1 dan model 3 memiliki nilai DIC (penalized deviance) dan WAIC yang mirip sedangkan model 2 memiliki DIC dan WAIC yang sedikit lebih tinggi. Model 3 memiliki nilai WAIC yang lebih rendah. Semakin rendah nilai WAIC, semakin baik kualitas model tersebut. Posterior predictive p-value ketiga model ini mendekati 0.5, berarti hasil prediksi model-model tersebut menyerupai data asli dan tidak bias (terlalu tinggi/rendah). Dari ketiga model, model ke-3 memiliki posterior predictive p-value terbaik.

V. Results

Dari semua model yang telah dibuat, model ketiga memiliki performa paling baik secara keseluruhan dalam memprediksi kualitas wine yang kita inginkan. Posterior summary dari semua parameter dapat dilihat pada Tabel 4.

Berdasarkan hasil dari SSVS, hanya terdapat beberapa variabel saja yang memiliki pengaruh terhadap kualitas dari wine, sehingga dapat disimpulkan bahwa untuk meningkatkan kualitas dari suatu wine, maka disarankan untuk lebih memperhatikan kadar asam asetat (atau asam lainnya yang mudah menguap) yang memberikan aroma pada wine (volatile acidity), jumlah gula yang tersisa setelah proses fermentasi selesai yang mempengaruhi tingkat kemanisan wine (residual sugar), sulfur dioksida bebas yang berguna sebagai pengawet (free sulfur dioxide), total sulfur dioksida untuk pengawet juga (total sulfur dioxide), kadar sulfat (sulphates), dan kadar alkohol (alcohol).

Tabel 4 Posterior summary model ketiga

	Mean	2.5%	97.5%
Y Average	0.6329	0.6184	0.6472
β fixed.acidity	**0.0971**	**-0.0289**	**0.2214**
β volatile.acidity	-0.7308	-0.8190	-0.6404
β citric.acidity	-0.0834	-0.1560	-0.0095
β residual.sugar	0.4101	0.2560	0.5560
β chlorides	**-0.0336**	**-0.1040**	**0.0367**
β free.sulfur.dioxide	0.2927	0.2041	0.3786
β total.sulfur.dioxide	-0.4317	-0.5292	-0.3335
β density	**-0.1625**	**-0.3780**	**0.0659**
β pH	0.1083	0.0167	0.2002
β sulphates	0.3154	0.2385	0.3921
β alcohol	1.0437	0.9125	1.1804

*tabel di atas menggunakan 95% credible interval, dimana variabel yang dibold adalah variabel yang tidak signifikan (range 95% CI mengandung 0).

R Code

Data Preprocessing

```
df <- read.csv("winequalityN.csv")
head(df)
```

```
summary(df)
```

```
df$fixed.acidity[is.na(df$fixed.acidity)] <- median(df$fixed.acidity, na.rm = TRUE)
df$volatile.acidity[is.na(df$volatile.acidity)] <- median(df$volatile.acidity, na.rm = TRUE)
df$citric.acid[is.na(df$citric.acid)] <- median(df$citric.acid, na.rm = TRUE)
df$residual.sugar[is.na(df$residual.sugar)] <- median(df$residual.sugar, na.rm = TRUE)
df$chlorides[is.na(df$chlorides)] <- median(df$chlorides, na.rm = TRUE)
df$pH[is.na(df$pH)] <- median(df$pH, na.rm = TRUE)
df$sulphates[is.na(df$sulphates)] <- median(df$sulphates, na.rm = TRUE)
```

```
summary(df)
```

```
df$quality <- ifelse(df$quality <= 5, 0, 1)
df
```

```
Y<-df[,13]
X<-scale(df[2:12])
n<-length(Y)
```

```
library(rjags)
burn <- 1000
iters <- 5000
chains <- 2
```

Model 1

##Model definition

```
mod_select <- textConnection("model{
  #Likelihood
  for(i in 1:n){
    Y[i] ~ dbern(q[i])
    logit(q[i]) <- alpha + inprod(X[i,],beta[])
    # WAIC Computation
    like[i] <- dbin(Y[i], q[i], 1)
  }

  #Prior
  for(j in 1:11){beta[j] ~ dnorm(0,0.01)}

  alpha ~ dnorm(0,0.01)

  #Posterior predictive checks
  for(i in 1:n){
    Y2[i] ~ dbern(q[i])
  }

  S <- sum(Y2[])/n

}")



##Generate samples with MCMC sampling for the Model

data <- list(Y=Y,X=X,n=n)
model_select <- jags.model(mod_select,data=data, n.chains = chains, quiet=TRUE)
update(model_select, burn)
samps_beta <- coda.samples(model_select,variable.names=c("S", "beta"), n.iter = iters, n.thin=5)


## Summary

summary(samps_beta)


## Convergence Check for the Model(Geweke)

# a |Z| > 2 indicates poor convergence
geweke.diag(samps_beta)
```

```
## Convergence Check for the Model (Gelman Rubin)

# 1 is good, >1.1 indicates poor convergence
gelman.diag(samps_beta)


## Auto-Correlation for the Model (plot)

autocorr.plot(samps_beta)


## Auto-Correlation for the Model

autocorr(samps_beta)


## Posterior Predictive Check for the Model

# take samples from the second chain
S <- samps_beta[[2]][,1]

#compute the test stats for the data
S0 <- (sum(Y)/n)
Snames <- "Proportion Y"

#compute the test stats for the model
pval <- rep(0,1)
names(pval) <- Snames

plot(density(S),xlim=range(c(S0,S)), xlab="S", ylab="Posterior probability",main=Snames)
abline(v=S0,col=2)
legend("topleft",c("Model","Data"),lty=1,col=1:2,bty="n")

pval <- mean(S>S0)

pval


## Model Evaluation for the Model (DIC)

DIC <- dic.samples(model_select, n.iter=iters,n.thin = 5)
DIC
```

Model Evaluation for the Model (WAIC)

```
samps_like_select <- coda.samples(model_select, variable.names = c("like"), n.iter=iters)
like_select <- rbind(samps_like_select[[1]], samps_like_select[[2]]) # Combine samples from the two chain
fbar <- colMeans(like_select)
P <- sum(apply(log(like_select),2,var))
WAIC <- -2*sum(log(fbar))+2*P
WAIC
```

SSVS & Model 2

SSVS Model

```
mod_SSVS <- textConnection("model{
  #Likelihood
  for(i in 1:n){
    Y[i] ~ dbern(q[i])
    logit(q[i]) <- alpha + inprod(X[i,],beta[])
  }

  #Prior
  for(j in 1:11){
    beta[j] <- gamma[j]*delta[j]
    gamma[j] ~ dbern(0.5)
    delta[j] ~ dnorm(0,0.01)
  }
  alpha ~ dnorm(0,0.01)
}")
```

SSVS MCMC Sampling

```
data <- list(Y=Y,X=X,n=n)
model_SSVS <- jags.model(mod_SSVS,data=data, n.chains = chains, quiet=TRUE)
update(model_SSVS, burn)
samps_SSVS <- coda.samples(model_SSVS,variable.names=c("beta"), n.iter = iters, n.thin=5)
```

SSVS Sample Summary

```
summary(samps_SSVS)
```

SSVS sample trace

```
plot(samps_SSVS)
```

inc prob

```
beta <- NULL
for(l in 1:chains){
  beta <- rbind(beta,samps_SSVS[[l]])
}

inc_prob <- apply(beta!=0,2,mean)
q <- t(apply(beta,2,quantile,c(0.5,0.05,0.95)))
out <- cbind(inc_prob,q)
out
```

Based on SSVS conducted, there are some features that are not really suitable for the model, therefore we create a new model with the selected features

Model with selected features (Model 2)
Selected features (Decide Y/target variables & X/features used in model with selected features)

```
# mengambil kolom ke-13 (quality) sebagai output variable dan kolom 2-12 sebagai input variables
Y<-df[,13]
X<-scale(df[, c(2, 4, 6, 7, 10, 11)])
n<-length(Y)
```

Selected Features Model

```
mod_select <- textConnection("model{
  #Likelihood
  for(i in 1:n){
    Y[i] ~ dbern(q[i])
    logit(q[i]) <- alpha + inprod(X[i,],beta[])
    # WAIC Computation
    like[i] <- dbin(Y[i], q[i], 1)
  }

  #Prior
```

```
  for(j in 1:6){beta[j] ~ dnorm(0,0.01)}

  alpha ~ dnorm(0,0.01)

  #Posterior predictive checks
  for(i in 1:n){
    Y2[i] ~ dbern(q[i])
  }

  S <- sum(Y2[])/n

}")
```

Generate samples with MCMC sampling for the Model with Selected Features (Beta)

```
data <- list(Y=Y,X=X,n=n)
model_select <- jags.model(mod_select,data=data, n.chains = chains, quiet=TRUE)
update(model_select, burn)
samps_beta_selected <- coda.samples(model_select,variable.names=c("S", "beta"), n.iter = iters,
n.thin=5)
```

```
summary(samps_beta_selected)
```

Sample Trace for the Model with Selected Features

```
par(mar = c(1, 1, 1, 1))
plot(samps_beta_selected)
```

Convergence Check for the Model with Selected Features (Geweke)

```
# a |Z| > 2 indicates poor convergence
geweke.diag(samps_beta_selected)
```

Convergence Check for the Model with Selected Features (Gelman Rubin)

```
# 1 is good, >1.1 indicates poor convergence
gelman.diag(samps_beta_selected)
```

Auto-Correlation for the Model with Selected Features (plot)

```
autocorr.plot(samps_beta_selected)
```

Auto-Correlation for the Model with Selected Features

```
autocorr(samps_beta_selected)
```

The autocorrelation of the samples are low, showing that the samples of each iterations are independent of each other

Posterior Predictive Check for the Model with Selected Features

```
# take samples from the second chain
S <- samps_beta_selected[[2]][,1]

#compute the test stats for the data
S0 <- (sum(Y)/n)
Snames <- "Proportion Y"

#compute the test stats for the model
pval <- rep(0,1)
names(pval) <- Snames

plot(density(S),xlim=range(c(S0,S)), xlab="S", ylab="Posterior probability",main=Snames)
abline(v=S0,col=2)
legend("topleft",c("Model","Data"),lty=1,col=1:2,bty="n")

pval <- mean(S>S0)

pval
```

Model Evaluation for the Model with Selected Features (DIC)

```
DIC <- dic.samples(model_select, n.iter=iters,n.thin = 5)
DIC
```

Model Evaluation for the Model with Selected Features (WAIC)

```
samps_like_select <- coda.samples(model_select, variable.names = c("like"), n.iter=iters)
like_select <- rbind(samps_like_select[[1]], samps_like_select[[2]]) # Combine samples from the two chain
fbar <- colMeans(like_select)
P <- sum(apply(log(like_select),2,var))
WAIC <- -2*sum(log(fbar))+2*P
WAIC
```

Model 3

Decide Y/target variables & X/features used in model 3

```
# mengambil kolom ke-13 (quality) sebagai output variable dan kolom 2-12 sebagai input variables
Y<-df[,13]
X<-scale(df[2:12])
n<-length(Y)
```

Model 3

```
mod <- textConnection("model{
  #Likelihood
  for(i in 1:n){
    Y[i] ~ dbern(q[i])
    logit(q[i]) <- alpha + inprod(X[i,],beta[])
    # WAIC Computation
    like[i] <- dbin(Y[i], q[i], 1)
  }

  #Prior
  for(j in 1:11){beta[j] ~ dnorm(0, 1)}

  alpha ~ dnorm(0, 1)

  #Posterior predictive checks
  for(i in 1:n){
    Y2[i] ~ dbern(q[i])
  }

  D <- sum(Y2[])/n

}")
```

Generate samples with MCMC sampling (Beta)

```
data <- list(Y=Y,X=X,n=n)
model <- jags.model(mod,data=data, n.chains = chains, quiet=TRUE)
update(model, burn)
samps_beta <- coda.samples(model,variable.names=c("D", "beta"), n.iter = iters, n.thin=5)
```

```
summary(samps_beta)
```

Sample Trace

```
par(mar = c(1, 1, 1, 1))
plot(samps_beta)
```

Convergence Check (Geweke)

```
# a |Z| > 2 indicates poor convergence
geweke.diag(samps_beta)
```

Convergence Check (Gelman Rubin)

```
# 1 is good, >1.1 indicates poor convergence
gelman.diag(samps_beta, multivariate=FALSE)
```

Auto-Correlation (plot)

```
autocorr.plot(samps_beta)
```

Auto-Correlation

```
autocorr(samps_beta)
```

The autocorrelation of the samples are low, showing that the samples of each iterations are independent of each other

Posterior Predictive Check

```
# take samples from the second chain
D <- samps_beta[[2]][,1]

#compute the test stats for the data
D0 <- (sum(Y)/n)
Dnames <- "Proportion Y"

#compute the test stats for the model
pval <- rep(0,1)
names(pval) <- Dnames

plot(density(D),xlim=range(c(D0,D)), xlab="D", ylab="Posterior probability",main=Dnames)
abline(v=D0,col=2)
legend("topleft",c("Model","Data"),lty=1,col=1:2,bty="n")

pval <- mean(D>D0)

pval
```

Model Evaluation (DIC)

```
DIC <- dic.samples(model,n.iter=iters,n.thin = 5)
DIC
```

Model Evaluation (WAIC)

```
samps_like <- coda.samples(model, variable.names = c("like"), n.iter=iters)
like <- rbind(samps_like[[1]], samps_like[[2]]) # Combine samples from the two chain
fbar <- colMeans(like)
P <- sum(apply(log(like),2,var))
WAIC <- -2*sum(log(fbar))+2*P
WAIC
```