STAT 656: Bayesian Data Analysis
Fall 2024
Homework 1

Juanwu Lu*

Synthetic Data

The *autoregressive model* is frequently used to analyze time series data. The simplest autoregressive model has order 1, and is abbreviated as AR(1). This model assumes that an observation y_i at time point i ($i = 1, \ldots, n$) is generated according to

$$y_i = \rho y_{i-1} + \epsilon_i,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ independently, and rho and σ are unknown parameters. For simplicity, we shall assume that y_0 is a fixed constant. We will also assume $|\rho| < 1$.

1. (5 points) **Solution:**

   Given the formulation above, the log-likelihood function is calculated as follows:

$$\begin{aligned}
\log L(\rho, \sigma^2 | y_0, y_1, \ldots, y_n) &= \log \prod_{i=1}^{n} (2\pi\sigma^2)^{-\frac{1}{2}} \cdot \exp\left\{ -\frac{(y_i - \rho y_{i-1})^2}{2\sigma^2} \right\} \\
&= -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \rho y_{i-1})^2 \\
&= -\frac{n}{2} \log(2\pi) - n \log(\sigma) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \rho y_{i-1})^2.
\end{aligned}$$

2. (10 points) **Solution:**

   The code implementation is as follows:

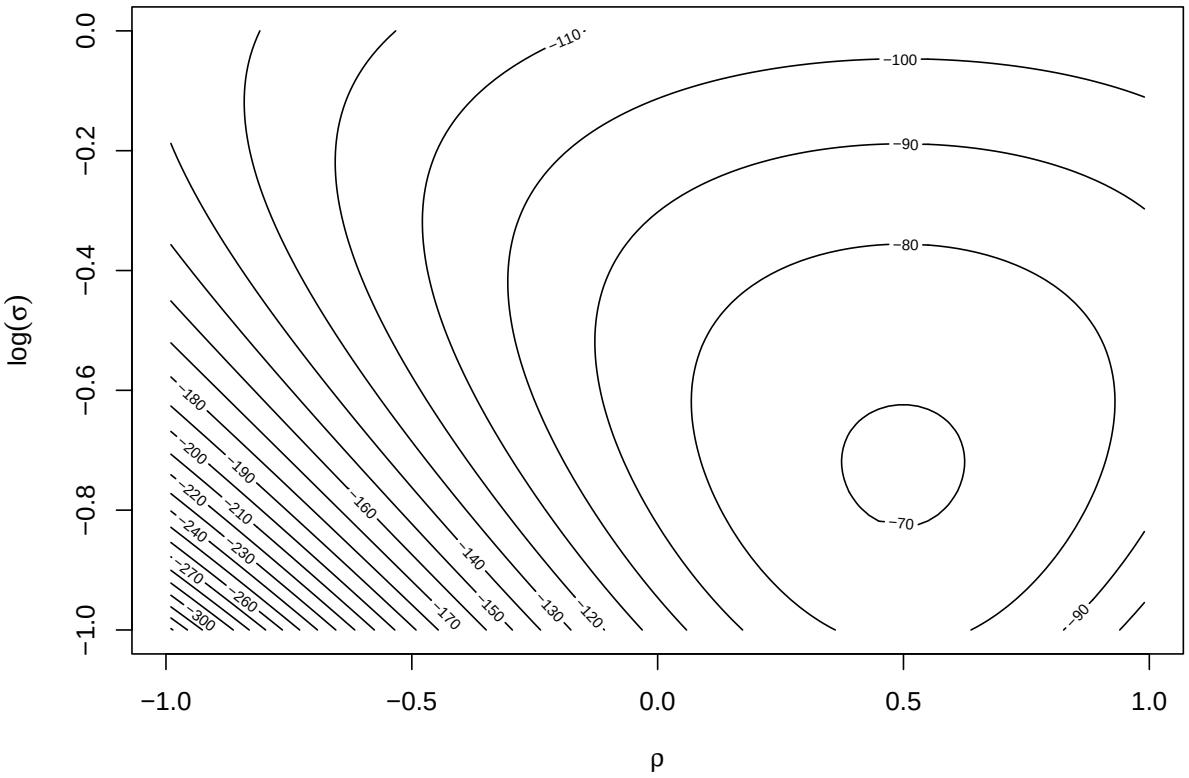
```
# Read data
if (file.exists("computation_data_hw_1.csv")) {
    data <- read.csv("computation_data_hw_1.csv")
    y <- data[['x']]
} else {
    stop("Cannot find the data file 'computation_data_hw_1.csv' at ", getwd())
}

# Define the log-likelihood function
ar_loglik <- function(rho, log_sig) {
    n <- length(y)
    sig <- exp(log_sig)
    rho <- rep(as.numeric(rho), times = n - 1)
    log_lik <- -0.5 * (
```

*College of Engineering, Purdue University, West Lafayette, IN, USA

```
        n * log(2 * pi)
        + 2 * n * log_sig
        + (y[1]^2 + sum((y[2:n] - rho * y[1:(n-1)])^2)) / sig^2
    )
    return(log_lik)
}

# Visualization
rho <- seq(-0.99, 0.99, length=100)
log_sig <- seq(-1.0, 0.0, length=100)
loglik <- outer(rho, log_sig, Vectorize(ar_loglik))
contour(
    x=rho,
    y=log_sig,
    z=loglik,
    xlab=expression(rho),
    ylab=expression(log(sigma)),
    nlevels=20,
)
```



3. (10 points) **Solution:**

Since the prior is independent, we have $p(\rho, \log(\sigma)) = p(\rho)p(\log(\sigma)) = \frac{1}{2} \cdot \frac{1}{10\sqrt{2\pi}} \exp\left\{-\frac{\log(\sigma)^2}{200}\right\}$. Therefore, the posterior density is proportional to the product of the likelihood and the prior, and then the log of the posterior density is calculated up to a constant as follows:

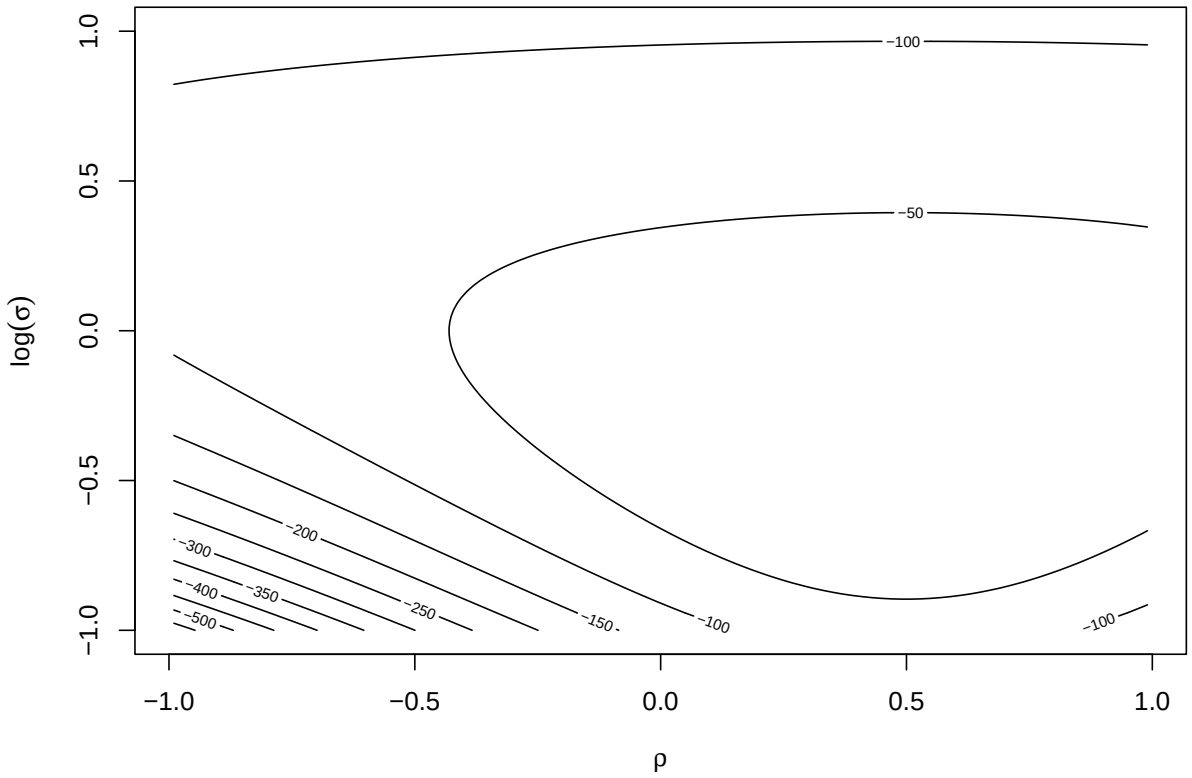
$$\begin{aligned}\log p(\rho, \log(\sigma)|y_0, y_1, \ldots, y_n) &= \log L(\rho, \log(\sigma)|y_0, y_1, \ldots, y_n) + \log p(\rho, \log(\sigma)) + C \\ &= -n\log(\sigma) - \frac{1}{2\exp(2 * \log(\sigma))} \sum_{i=1}^{n} (y_i - \rho y_{i-1})^2 - \frac{\log(\sigma)^2}{200} + C'\end{aligned}$$

where C is the constant log-normalizer and C' is a constant irrelevant to ρ and $\log(\sigma)$. The visualization of the log of the posterior density is as follows:

```r
# Read data
if (file.exists("computation_data_hw_1.csv")) {
    data <- read.csv("computation_data_hw_1.csv")
    y <- data[['x']]
} else {
    stop("Cannot find the data file 'computation_data_hw_1.csv' at ", getwd())
}

# Define the log-likelihood function
ar_logpost <- function(rho, log_sig) {
    n <- length(y)
    sig <- exp(log_sig)
    rho <- rep(as.numeric(rho), times = n - 1)
    log_post <- -(
        n * log_sig
        + (y[1]^2 + sum((y[2:n] - rho * y[1:(n-1)])^2)) / sig^2
        + log_sig^2 / 200
    )
    return(log_post)
}

# Visualization
rho <- seq(-0.99, 0.99, length=100)
log_sig <- seq(-1.0, 1.0, length=100)
logpost <- outer(rho, log_sig, Vectorize(ar_logpost))
contour(
    x=rho,
    y=log_sig,
    z=logpost,
    xlab=expression(rho),
    ylab=expression(log(sigma)),
    nlevels=20,
)
```



Compared to the log-likelihood function, the log posterior density is more concentrated around the maximum likelihood estimate of $(\rho, \log(\sigma))^\intercal$. The prior is not overly informative since the shape of the posterior density is still similar to the likelihood function, indicating that the posterior is mainly determined by the likelihood function.

4. (10 points) Draw 1000 values of $(\rho, \log(\sigma))^\intercal$ from a discrete grid approximation to the posterior. Be sure to describe your choice of discrete grid. Hint: The previous step can be helpful in this regard, together with the R function sample. Again, look at the code associated with the lectures.

5. (5 points) Use these draws to calculate the following summaries for each of ρ and $\log(\sigma)$: $0.025, 0.25, 0.5, 0.75, 0.975$ quantiles, mean, standard deviation, skewness, and kurtosis. You can use the library `moments` if you want.

6. (10 points) Write an `R` function that takes parameters $(\rho, \log(\sigma))^\intercal$ and simulates a new dataset y^{rep} according to the AR process. Recall that we can simulate from the posterior predictive distribution of new datasets y^{rep} given todays dataset as follows: first simulate a parameter set from the posterior distribution, and use this to simulate a new dataset. Use your two earlier `R` functions to generate 1000 such posterior predictive samples. Summarize your draws.

7. (10 points) Compare the observed data to these posterior predictive summaries. Also create a plot where the observed data is suporposed on these posterior predictive trajectories. What can you say about the model fit? Does the model appear appropriate?