Original Research Paper

Volume - 12 | Issue - 02 | February - 2022 | PRINT ISSN No. 2249 - 555X | DOI : 10.36106/ijar

Statistics

A SIMPLE GIBBS SAMPLER FOR THE STATE ESTIMATION IN WIRELESS COMMUNICATIONS

Abhirup Banerjee Department of Statistics, University Of Calcutta, India.

Kiranmoy Das* Applied Statistics Division, Indian Statistical Institute, Kolkata, India. *Corresponding Author

ABSTRACT In the recent years, wireless communications are extremely useful in many disciplines including health monitoring, environment monitoring, signal processing etc. State estimation and prediction are quite challenging tasks in wireless communications. Traditionally, in the literature, dynamic state-space models have been used for the state estimation and predic- tion purpose. The estimation method is based on Kalman-Filter which is computationally demanding. In this work, we consider computationally simpler Gibbs sampler algorithm for the state estimation. We consider three different cases, (i) continuous state values, (ii) binary (0/1) state values, and (iii) categorical state values with more than two categories. We consider a simple linear model for the prediction purpose, and the underlying regression coefficients are estimated by Gibbs sampler. We compute the misclassification proportions for assessing the practical usefulness of our estimation approach. A real dataset where 200 wireless sensor nodes are used for measuring the temperature of a chamber is analysed in this work.

KEYWORDS : Gibbs Sampler, Longitudinal Data, MCMC, Model Selection, Wireless Sensor Network.

1 INTRODUCTION

In the last decade, we have seen a revolution in technology, and in data science, and be- cause of that wireless communication has become an extremely useful topic for research and development. A vast amount of data can be collected using wireless sensors without human- to-human interactions, and the collected data can be automatically sent to the base station for the analysis and for the prediction purpose. Nagarajan et al. (2009), Chatterjee et al. (2020) used wireless communications for providing better health services to the citizens in smart cities. Winkler et al. (2012) proposed models for wireless communications in military surveillance. Wireless sensor nodes, which are low-powered tiny devices are also used in geology (Shi-Young et al. 2012), and in criminology (Gong et al. 2016).

Using such useful sensor nodes, huge amount of data can be collected in a relatively short time, and hence an efficient data analysis technique is required. Traditional state- space models are typically used for analysing such datasets, where Kalman-Filter is used for the estimation purpose (Chatterjee and Das 2018). However, an alternative Bayesian approach is also proposed in Chatterjee et al. (2017) where Markov Chain Monte Carlo (MCMC) is used for the estimation purpose. Chatterjee et al. (2017) showed that an MCMC based computation is not only computationally efficient, but provides more accurate results than the traditional Maximum Likelihood Estimates (MLE). In fact, Chatterjee et al. (2016) proposed a non-parametric Bayesian approach for the state estimation and anomaly detection in a cluster-based wireless sensor network. Bayesian methods are typically preferred because they can incorporate the relevant prior information on the regression coefficients, and can update the estimates based on the available data in a dynamic way.

In a traditional linear model with continuous state values, Bayesian computations are quite straight-forward. One needs to specify the prior distributions for the regression coef- ficients, and then explicitly write down the likelihood function for the complete data. The posterior distribution, which is proportional to the product of the likelihood and the priordistributions, is used for computing the full conditional distributions for the regression coef- ficients. Based on these full conditional distributions, one can simply run a Gibbs sampler or a Metropolis-Hastings algorithm for estimating the regression coefficients. Chatterjee and Venkateswaran (2015) developed such computational approach for time synchronization in wireless communications.

However, for the binary or categorical state values, a generalized linear model is used. Albert and Chib (1993) showed that for such models posterior distributions are intractable. For addressing this issue, they proposed a data-augmentation technique where a latent continuous random variable is used for the computation purpose. The latent variable is connected to the observed categorical state values through some known thresholds. This data-augmentation technique has been widely used in the Bayesian literature. Biswas and Das (2020), Biswas et al. (2020), Biswas and Das (2021), Bhuyan et al. (2019) used this approach for modeling continuous longitudinal data containing excess zeros. More recently, Chatterjee et al. (2020) used this technique for automated health monitoring using discrete- time wireless sensors. In our work, we exploit this algorithm for analysing and predicting the binary and other categorical state values.

Our current work is motivated by a dataset where 200 sensor nodes are used for measur- ing the temperature of a chamber. The temperature of the chamber increases with time, and these sensors measure it (as state values) at five discrete time points. The initial temperature was at zero, and then it increases upto 800 and above. A plot of the raw dataset is shown in Figure 1. Note that these 200 sensors form a network, i.e. they share information over time. We use a linear statistical model proposed by Chatterjee et al. (2017) for handling such dependence among the sensor nodes. Then we fit the mean curve, and predict the tem- perature at the intermediate time points. Next, we translate the continuous measurements into binary (0/1) values, since such a transformation makes the system energy-efficient. We use data from 160 sensor nodes, and predict the binary states for the 40 nodes, and compute the misclassification proportion. Then, we consider more than two categories, and develop

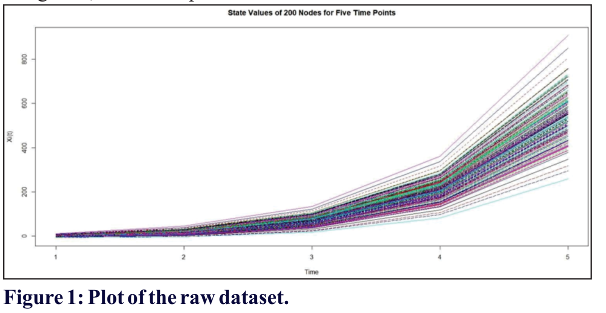

Figure 1: Plot of the raw dataset.

the Gibbs sampler algorithm for the similar estimation and prediction.

The rest of the paper is organized as follows. In Section 2, we analyse the continuous state values in a network framework, and estimate the regression coefficients. The Gibbs sampler algorithm is discussed in detail in this section. In Section 3, we consider the binary, and the categorical states (for more than two categories), and develop the corresponding computational algorithm. The effectiveness of the model is discussed in this section. Finally, in Section 4 we make some concluding remarks.

Table 1: AIC/BIC for different orders

Order of $f(t)$	AIC	BIC
1	5301.99	5326.53
2	5078.07	5107.52
3	5079.37	5113.73

2 Model and Estimation Method

2.1 Linear Regression Model

In this section, we consider the linear regression model proposed in

Chatterjee et al. (2017) for predicting the state values of the sensor nodes. The model is as follows:

$$X_i(t) = f(t) + \alpha X_i(t-1) + \beta Z_{-i}(t-1) + e_i(t), \quad (1)$$

where $X_i(t)$ denotes the state value of the i-th sensor node at time t, for $i = 1, 2, \ldots, 200$, and $t = 1, 2, \ldots, 5$. Here, f denotes the general effect of time on the current state values, $X_i(t-1)$ is the immediate predecessor state value, and α denotes the effect of $X_i(t-1)$ on $X_i(t)$. Additionally, $Z_{-i}(t-1)$ denotes the average state values of all but the i-th state value at time $t-1$, and the corresponding regression coefficient β is known as "neighbourhood effect". The random errors *ei(t)* are identically and independently distributed as $N(0, \sigma^2)$.

The general effect of time f can be modeled by P-splines (Chatterjee et al. 2016), or by Orthogonal Legendre polynomials (Biswas and Das 2020). However, for the sake of simplicity, we model this using a polynomial fucntion of time. In other words, we consider, $f(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_r t^r$, where the optimal order r can be obtained using the information criteria (AIC/BIC). For the given dataset, we consider r=1,2,3; and the results are summarized in Table 1.

Thus, we see that the AIC and BIC are smallest for r=2.

So, our model becomes: $X_i(t) = a_0 + a_1 t + a_2 t^2 + \alpha X_i(t-1) + \beta Z_{-i}(t-1) + i(t)$.

2.2 Gibbs Sampler for Estimation

We develop a Bayesian estimation approach for this analysis. We consider the following prior distributions for the regression coefficients:

$$a_j \sim N(\mu_j, \sigma^2), j = 0, 1, \ldots, r; \alpha \sim N(\mu_\alpha, \sigma^2); \quad \beta \sim N(\mu_\beta, \sigma^2);$$
$$\sigma_e^2 \sim IG(shape = v_1, rate = v_2), \quad (2)$$

where IG stands for an Inverse Gamma distribution.
The joint likelihood function can be written as follows:

where $E = \sum_{i=1}^{n} \sum_{t=2}^{T} [X_i(t) - f(t) - \alpha X_i(t-1) - \beta Z_i(t-1)]^2$.

The joint posterior distribution is, therefore, given as follows:

$$\pi(\boldsymbol{a}, \alpha, \beta, \sigma_\epsilon^2 \mid X) \propto L \times \prod_{j=0}^{r} P(a_j) \times P(\alpha) \times P(\beta) \times P(\sigma_\epsilon^2), \quad (3)$$

where π denotes the posterior distribution, and P denotes the prior distribution. Based on (3), the full conditional distributions are obtained as follows:

$$\text{(i) } \pi(a_0 \mid a_1, a_2, \alpha, \beta, \sigma_\epsilon^2, X) \sim N\left(\frac{T_1^{a_0} - T_2^{a_0} - T_3^{a_0} - T_4^{a_0} - T_5^{a_0} + T_6^{a_0}}{V_0}, \frac{1}{V_0}\right),$$

$$\text{where, } T_1^{a_0} = \frac{\sum_{i=1}^{n}\sum_{t=2}^{T} X_i(t)}{\sigma_\epsilon^2}, T_2^{a_0} = \frac{a_1 \sum_{i=1}^{n}\sum_{t=2}^{T} t}{\sigma_\epsilon^2}, T_3^{a_0} = \frac{a_2 \sum_{i=1}^{n}\sum_{t=2}^{T} t^2}{\sigma_\epsilon^2},$$

$$T_4^{a_0} = \frac{\alpha \sum_{i=1}^{n}\sum_{t=2}^{T} X_i(t-1)}{\sigma_\epsilon^2}, T_5^{a_0} = \frac{\beta \sum_{i=1}^{n}\sum_{t=2}^{T} Z_i(t-1)}{\sigma_\epsilon^2}, T_6^{a_0} = \frac{\mu_0}{\sigma_0^2}, V_0 = \left(\frac{\sum_{i=1}^{n}\sum_{t=2}^{T} 1}{\sigma_\epsilon^2} + \frac{1}{\sigma_0^2}\right)$$

$$\text{(ii) } \pi(a_1 \mid a_0, a_2, \alpha, \beta, \sigma_\epsilon^2, X) \sim N\left(\frac{T_1^{a_1} - T_2^{a_1} - T_3^{a_1} - T_4^{a_1} - T_5^{a_1} + T_6^{a_1}}{V_1}, \frac{1}{V_1}\right),$$

$$\text{where, } T_1^{a_1} = \frac{\sum_{i=1}^{n}\sum_{t=2}^{T} tX_i(t)}{\sigma_\epsilon^2}, T_2^{a_1} = \frac{a_0 \sum_{i=1}^{n}\sum_{t=2}^{T} t}{\sigma_\epsilon^2}, T_3^{a_1} = \frac{a_2 \sum_{i=1}^{n}\sum_{t=2}^{T} t^3}{\sigma_\epsilon^2},$$

$$T_4^{a_1} = \frac{\alpha \sum_{i=1}^{n}\sum_{t=2}^{T} tX_i(t-1)}{\sigma_\epsilon^2}, T_5^{a_1} = \frac{\beta \sum_{i=1}^{n}\sum_{t=2}^{T} tZ_i(t-1)}{\sigma_\epsilon^2}, T_6^{a_1} = \frac{\mu_1}{\sigma_1^2}, V_1 = \left(\frac{\sum_{i=1}^{n}\sum_{t=2}^{T} t^2}{\sigma_\epsilon^2} + \frac{1}{\sigma_1^2}\right).$$

$$\text{(iii) } \pi(a_2 \mid a_1, a_0, \alpha, \beta, \sigma_\epsilon^2, X) \sim N\left(\frac{T_1^{a_2} - T_2^{a_2} - T_3^{a_2} - T_4^{a_2} - T_5^{a_2} + T_6^{a_2}}{V_2}, \frac{1}{V_2}\right),$$

$$\text{where, } T_1^{a_2} = \frac{\sum_{i=1}^{n}\sum_{t=2}^{T} t^2X_i(t)}{\sigma_\epsilon^2}, T_2^{a_2} = \frac{a_0 \sum_{i=1}^{n}\sum_{t=2}^{T} t^2}{\sigma_\epsilon^2}, T_3^{a_2} = \frac{a_1 \sum_{i=1}^{n}\sum_{t=2}^{T} t^3}{\sigma_\epsilon^2},$$

$$T_4^{a_2} = \frac{\alpha \sum_{i=1}^{n}\sum_{t=2}^{T} t^2X_i(t-1)}{\sigma_\epsilon^2}, T_5^{a_2} = \frac{\beta \sum_{i=1}^{n}\sum_{t=2}^{T} t^2Z_i(t-1)}{\sigma_\epsilon^2}, T_6^{a_2} = \frac{\mu_2}{\sigma_2^2}, V_2 = \left(\frac{\sum_{i=1}^{n}\sum_{t=2}^{T} t^4}{\sigma_\epsilon^2} + \frac{1}{\sigma_2^2}\right)$$

$$\text{(iv) } \pi(\alpha \mid a_0, a_1, a_2, \beta, \sigma_\epsilon^2, X) \sim N\left(\frac{T_1^{\alpha} - T_2^{\alpha} - T_3^{\alpha} + T_4^{\alpha}}{V_\alpha}, \frac{1}{V_\alpha}\right),$$

$$\text{where, } T_1^{\alpha} = \frac{\sum_{i=1}^{n}\sum_{t=2}^{T} X_i(t)X_i(t-1)}{\sigma_\epsilon^2}, T_2^{\alpha} = \frac{\sum_{i=1}^{n}\sum_{t=2}^{T} X_i(t-1)(a_0 + a_1 + a_2t^2)}{\sigma_\epsilon^2},$$

$$T_3^{\alpha} = \frac{\beta \sum_{i=1}^{n}\sum_{t=2}^{T} X_i(t-1)Z_i(t-1)}{\sigma_\epsilon^2}, T_4^{\alpha} = \frac{\mu_\alpha}{\sigma_\alpha^2}, V_\alpha = \left(\frac{\sum_{i=1}^{n}\sum_{t=2}^{T} X_i^2(t-1)}{\sigma_\epsilon^2} + \frac{1}{\sigma_\alpha^2}\right).$$

For the current analysis, we use the following prior specifications:

$$a_0 \sim N(\mu_0 = 0, \sigma_0^2 = 1000), \ a_1 \sim N(\mu_1 = 0, \sigma_1^2 = 1000), \ a_2 \sim N(\mu_2 = 0, \sigma_2^2 = 1000);$$
$$\alpha \sim N(\mu_\alpha = 0, \sigma_\alpha^2 = 1000), \ \beta \sim N(\mu_\beta = 0, \sigma_\beta^2 = 1000), \ \sigma_\varrho^2 \sim IG(v_1 = 0.01, v_2 = 0.01).$$

We note that our prior distributions are mostly non-informative, and have minimal effects on the final estimates as observed in a sensitivity analysis (results not shown).

We run the Gibbs Sampling technique to generate from the respective full conditionals. We run 10^6 iterations and remove the first 50,000 values as "burn-in". This removes the effects of the initial specifications of the regression coefficients. Then, we save every 1000-th observation to thin the Markov Chain. This removes the auto-correlation among the MCMC iterations. Regression coefficients are estimated as their respective marginal posterior sample means.

2.3 RESULTS

In Table 2, we summarize the estimated regression coefficients, and their respective 95% Bayesian credible intervals based on MCMC iterations. We note that except the "neigh- bourhood effect" (β), for all other coefficients the credible intervals do not contain zeros. This reflects the absence of neighbourhood effect in this dataset, but the effect of immediate predecessor value is significant.

Next, we plot the estimated mean state curve along with the observed data. We see that the estimated curve passes through the central part of the data, and thus the estimation accuracy of the MCMC based approach is established.

Table 2: Estimates and Credible Intervals for the Coefficients

Coefficient	Estimate	95% CI
a_0	3.45	(1.23,4.68)
a_1	1.91	(0.67,3.19)
a_2	2.33	(1.15,4.35)
α	2.31	(0.89,3.24)
β	-0.0075	(-1.18,0.87)

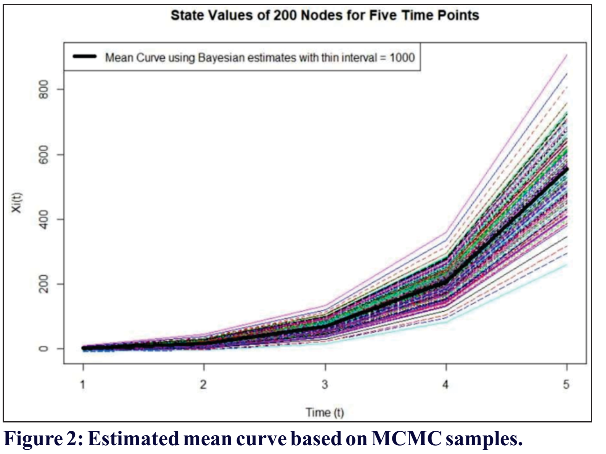

Figure 2: Estimated mean curve based on MCMC samples.

3. Categorical State Values

3.1 Binary State

Next, we consider binary state values for the given dataset. As mentioned in Chatterjee et al. (2020), binary state values can be stored and sent with low battery power, and hence it is energy efficient. Additionally, for some specific applications (e.g. health monitoring), binary states are more meaningful to the non-experts. For example, a non-medical person cannot understand whether 135 mEq/L sodium level in blood is a good sign or not. However, a binary state (normal/abnormal) is more meaningful to everyone.

3.1.1 Introducing Latent Variables

For the above dataset, we consider the algorithm proposed in Albert and Chib (1993). We introduce latent variables X (t) as follows:

$$X_i(t) = \begin{cases} 1, & for \ X_i^*(t) > \delta \\ 0, & for \ X_i^*(t) \le \delta, \end{cases}$$

where the threshold δ is known. For our case, we consider δ=150.

The latent state values $X^*_i(t)$ are modeled similar to Section 2. The linear regression model used for this purpose is the following:

$$X_i^*(t) = a_0 + a_1 t + a_2 t^2 + \alpha X_i^*(t-1) + \beta Z_{-i}^*(t-1) + \epsilon_i(t),$$

Since at the first time point there is no data for the previous times, we model:

3.1.2 Prior Specification and Gibbs Sampler

We use the following prior distributions :

$a_0 \sim N(\mu_0 = 0, \sigma_0^2 = 1000)$, $a_1 \sim N(\mu_1 = 0, \sigma_1^2 = 1000)$, $a_2 \sim N(\mu_2 = 0, \sigma_2^2 = 1000)$,

$\alpha \sim N(\mu_\alpha = 0, \sigma_\alpha^2 = 1000)$, $\beta \sim N(\mu_\beta = 0, \sigma_\beta^2 = 1000)$, $\sigma_\epsilon^2 \sim IG(\nu_1 = 0.01, \nu_2 = 0.01)$.

Suppose the initial estimates of the regression coefficients (which are obtained by sampling from the respective prior densities) are $a_0^{(0)}, a_1^{(0)}, a_2^{(0)}, \alpha^{(0)}, \beta^{(0)}, \sigma_\epsilon^{2\,(0)}$.

Suppose, the initial estimates are $a^{(0)}, a^{(0)}, a^{(0)}, \alpha^{(0)}, \beta^{(0)}, \sigma^{2\,(0)}$, which are sampled from the respective prior distributions.

Step 1: Take j = 0.

If $X_i(1) = 1$, simulate $X_i^{*(j)}(1)$ from $N(a_0^{(j)}, 1)$, truncated at left by δ ;

If $X_i(1) = 0$, simulate $X_i^{*(j)}(1)$ from $N(a_0^{(j)}, 1)$, truncated at right by δ.

Step 2:

If $X_i(t) = 1$, simulate $X_i^{*(j)}(t)$ from $N(a_0^{(j)} + a_1^{(j)} t + a_2^{(j)} t^2 + \alpha^{(j)} X_i^{*(j)}(t-1) + \beta Z_{-i}^{*(j)}(t-1), 1)$, truncated at left by δ ;

If $X_i(t) = 0$, simulate $X_i^{*(j)}(t)$ from $N(a_0^{(j)} + a_1^{(j)} t + a_2^{(j)} t^2 + \alpha^{(j)} X_i^{*(j)}(t-1) + \beta Z_{-i}^{*(j)}(t-1), 1)$, truncated at right by δ.

Step 3:

Using the simulated $X_i^{*(j)}(t)$ values, we simulate $a_0^{(j+1)}, a_1^{(j+1)}, a_2^{(j+1)}, \alpha^{(j+1)}, \beta^{(j+1)}, \sigma_\epsilon^{2\,(j+1)}$ from their respective full conditional distributions, as derived in Section 2.2.

Step 4:

Repeat Steps 1, 2 and 3 for $j = 1, 2, \ldots, N$; where N is the simulation number. Consider, $N = 10^6$ for our study.

Table 3: Percentage of predicted class vs. actual class

	Predicted class=1	Predicted class=0
Actual class=1	98	2
Actual class=0	1.67	98.33

We run 10^6 iterations and remove the first 50,000 values for each parameter as burn-in and henceforth take every 1000-th observation to thin the Markov Chain. We calculate the respective sample means. Regression coefficients are estimated by their respective sample means.

3.1.3 RESULTS

Now, we use the estimated parameter values $\hat{a}_0$, $\hat{a}_1$, $\hat{a}_2$, $\hat{\alpha}$, $\hat{\beta}$ for predicting the test dataset.

Since, the test set contains only dichotomous state values (1 or 0), for the first time point we need to generate latent variables based on truncated normal distribution depending on $\hat{a}_0$ for every i as below:

$$X_i(1) = \begin{cases} 1, & simulate\ X_i^*(1) \sim N(\hat{a}_0, 1) I(Z \geq \delta), \\ 0, & simulate\ X_i^*(1) \sim N(\hat{a}_0, 1) I(Z < \delta). \end{cases}$$

For $t = 2, 3, \ldots, T$, and every i, we can predict the remaining $X_i^*(t)$ based on the parameter estimates and $X_i^*(1)$ as: $X_i^*(t) = \hat{a}_0 + \hat{a}_1 t + \hat{a}_2 t^2 + \hat{\alpha} X_i^*(t-1) + \hat{\beta} Z_{-i}^*(t-1)$.

We classify this predicted value as 1 or 0, based on its value above or below the threshold. If we predict 1 as 0, or 0 as 1, this is a misclassification. Table 3 summarizes the misclassifica- tion proportions. We note that more than 98% of the times, the predicted class match with the actual class. So, this result demonstrates the practical usefulness of the Gibbs sampler algorithm for binary state estimation.

3.2 More than Two-Category State

Now, we extend the binary states to more than two-category states. Sometimes, for a better understanding of the underlying process it is beneficial to consider several categories for the state. We note that categorical states can also be sent and stored with lower energy than the continuous states. In the context of health monitoring, Chatterjee et al. (2020) demonstrated that the health status of a patient is better described by categories (e.g. excellent, good, fair, not so good, critical) than just binary outcomes (normal/abnormal).

By considering appropriate cutoff points $\gamma_1, \gamma_2, \ldots, \gamma_r$, we link the observed categories to the unobserved latent state values as follows:

$$X_i(t) = \begin{cases} 1, & for\ X_i^*(t) \leq \gamma_1, \\ 2, & for\ \gamma_1 < X_i^*(t) \leq \gamma_2 \\ 3, & for\ \gamma_2 < X_i^*(t) \leq \gamma_3 \\ 4, & for\ X_i^*(t) > \gamma_3. \end{cases}$$

where $\gamma_1 = 50$, $\gamma_2 = 200$, $\gamma_3 = 400$.

Similar to the binary case, we model the latent continuous state values as follows:

$$X_i^*(t) = \hat{a}_0 + \hat{a}_1 t + \hat{a}_2 t^2 + \hat{\alpha} X_i^*(t-1) + \hat{\beta} Z_{-i}^*(t-1).$$

where the residuals $\epsilon_i(t)$ $N(0, \sigma^2)$, independently.

3.2.1 Prior Distributions and Gibbs Sampler

We use the following prior distributions for our analysis:

$a_0 \sim N(\mu_0 = 0, \sigma_0^2 = 1000)$, $a_1 \sim N(\mu_1 = 0, \sigma_1^2 = 1000)$, $a_2 \sim N(\mu_2 = 0, \sigma_2^2 = 1000)$,

$\alpha \sim N(\mu_\alpha = 0, \sigma_\alpha^2 = 1000)$, $\beta \sim N(\mu_\beta = 0, \sigma_\beta^2 = 1000)$, $\sigma_\varphi^2 \sim IG(\nu_1 = 0.01, \nu_2 = 0.01)$.

Suppose the initial estimates of the regression coefficients (which are obtained by sampling from the respective prior densities) are $a_0^{(0)}, a_1^{(0)}, a_2^{(0)}, \alpha^{(0)}, \beta^{(0)}, \sigma_\epsilon^{2\,(0)}$.

Step 0:

For every i, we simulate the latent variables $X_i(1)$ corresponding to the very first time point as follows:

$$X_i(1) = \begin{cases} 1, & simulate\ X_i^*(1) \sim N(a_0^{(0)}, 1) I(Z \leq 50), \\ 2, & simulate\ X_i^*(1) \sim N(a_0^{(0)}, 1) I(50 < Z \leq 200), \\ 3, & simulate\ X_i^*(1) \sim N(a_0^{(0)}, 1) I(200 < Z \leq 400), \\ 4, & simulate\ X_i^*(1) \sim N(a_0^{(0)}, 1) I(Z > 400). \end{cases}$$

Step 1:

For every i and for $t = 2, \ldots, T$, let us define: $f^{(j)}(t) = a_0^{(j)} + a_1^{(j)} t + a_2^{(j)} t^2 + \alpha^{(j)} X_i^{*(j)}(t-1) + \beta Z_{-i}^{*(j)}(t-1)$.

We simulate latent variables $X_{(j)}^*(t)$ based on $f^{(j)}(t)$ as follows:

$$X_i(t) = \begin{cases} 1, & simulate\ X_i^*(t) \sim N(f^{(j)}(t), 1) I(Z \leq 50), \\ 2, & simulate\ X_i^*(t) \sim N(f^{(j)}(t), 1) I(50 < Z \leq 200), \\ 3, & simulate\ X_i^*(t) \sim N(f^{(j)}(t), 1) I(200 < Z \leq 400), \\ 4, & simulate\ X_i^*(t) \sim N(f^{(j)}(t), 1) I(Z > 400). \end{cases}$$

Step 2:

Using the simulated $X^{*(j)}(t)$ values, we simulate updated estimates $a_0^{(j+1)}, a_1^{(j+1)}, a_2^{(j+1)}, \alpha^{(j+1)}, \beta^{(j+1)}, \sigma_\epsilon^{2\,(j+1)}$ from their respective full conditional distributions, as derived in Section 2.2.

Step 3:

Repeat the steps 1 and 2, for $j = 1, 2, \ldots, N$, where $N = 10^6$ in our analysis.

Step 4:

We run 10^6 iterations and remove the first 50,000 values for each parameter as burn-in and henceforth take every 1000-th observation to thin the Markov Chain. We calculate the respective sample means. Regression coefficients are estimated by their respective sample means.

Table 4: Percentage of predicted class vs. actual class

	Predicted class=1	Predicted class=2	Predicted class=3	Predicted class=4
Actual class=1	100	0	0	0
Actual class=2	0	100	0	0
Actual class=3	0	0	95	5
Actual class=4	0	0	5	95

3.2.2 Calculating Misclassifications

We use the estimated parameter values $\hat{a}_0$, $\hat{a}_1$, $\hat{a}_2$, $\hat{\alpha}$, $\hat{\beta}$ for predicting the states of the test dataset.

Since, the test set contains only categorical state values (1,2,3 or 4), for the first time point we need to generate latent variables based on truncated normal distribution depending on $\hat{a}_0$ for every i as follows:

$$X_i(1) = \begin{cases} 1, & simulate\ X_i^*(1) \sim N(\hat{a_0}, 1)I(Z \leq 50), \\ 2, & simulate\ X_i^*(1) \sim N(\hat{a_0}, 1)I(50 < Z \leq 200), \\ 3, & simulate\ X_i^*(1) \sim N(\hat{a_0}, 1)I(200 < Z \leq 400), \\ 4, & simulate\ X_i^*(1) \sim N(\hat{a_0}, 1)I(Z > 400). \end{cases}$$

For t = 2, 3, . . . , T and every i, we can predict the remaining $X_i^*(t)$ based on the param- eter estimates and Xi (1) as:

$$X_i^*(t) = \hat{a_0} + \hat{a_1}t + \hat{a_2}t^2 + \hat{\alpha}X_i^*(t-1) + \hat{\beta}Z_{-i}^*(t-1).$$

We can classify these predicted values as 1,2,3 or 4 based on the pre-specified thresholds.

Table 4 summarizes the results. We note that the categories 1 and 2 are predicted with 100% accuracy, while categories 3 and 4 are predicted with 95% accuracy. The misclassifi- cation proportion is 0.05, for categories 3 and 4. This illustrates that the proposed Gibbs sampler is quite accurate in classifying the categorical state values over time.

4. DISCUSSION

In this paper, we have analysed a real dataset collected by wireless sensor nodes. These sensor nodes form a network, and share information over time. We used linear regression model for analysing the continuous state values, and then we implement Bayesian data-augmentation technique for binary and categorical states with more than two categories. We use the computationally efficient Gibbs sampler for estimating the regression coefficients. For the continuous case, we plot the estimated mean curve which nicely summarizes the dataset. For the categorical states, we compute the misclassification proportion, and as demonstrated in Section 3, the proposed approach results in very small misclassification proportions. This shows that for analysing similar data collected by wireless sensor nodes, Gibbs sampler can be a very effective computational approach.

Our current dataset does not contain any missing data. However, in longitudinal studies it is not uncommon to come across missing observations at some time points. In particular, for wireless communication sometimes some sensor nodes are kept in sleep mode since in the sleep mode the sensors consume very less energy. As demonstrated in Chatterjee et al. (2017), all sensor nodes will not be active at all time point, but some will remain in the sleep mode. Under such situations, the linear regression model can be used for imputing the missing observations based on the observed states of the active sensors at that time points. A Bayesian joint model can simultaneously update the regression coefficients and the missing observations, and thus resulting in an efficient model for the state estimation. In the other longitudinal studies, if the underlying missingness is ignorable (e.g. missing at random), a similar data-augmentation technique can be implemented (Biswas and Das 2020). However, if the missingness is non-ignorable, then one can implement the methods discussed in Daniels and Hogan (2008). We leave these as our future research avenues.

REFERENCES

1. Albert, J.H., and Chib, S. (1993) Bayesian analysis of binary and polychotomous response data. Journal of the American statistical Association 88: 669–679.
2. Bhuyan, P., Biswas, J., Ghosh, P. and Das, K. (2019) A Bayesian two-stage regression ap- proach of analysing longitudinal outcomes with endogeneity and incompleteness. Sta- tistical Modelling 19: 157–173.
3. Biswas, J. and Das, K. (2020) A Bayesian approach of analysing semi-continuous longitudinal data with monotone missingness. Statistical Modelling 20: 148–170.
4. Biswas, J. and Das, K. (2021) A Bayesian quantile regression approach to multivariate semi- continuous longitudinal data. Computational Statistics 36: 241–260.
5. Biswas, J., Ghosh, P. and Das, K. (2020) A semi-parametric quantile regression approach to zero-inflated and incomplete longitudinal outcomes. AStA Advances in Statistical Analysis 104: 261–283.
6. Chatterjee, A. and Venkateswaran, P. (2015) An efficient statistical approach for time syn- chronization in wireless sensor networks. International Journal of Communication Sys- tems 29: 722–733.
7. Chatterjee, A., Venkateswaran, P. and Das, K. (2016) Simultaneous State Estimation of Cluster-Based Wireless Sensor Networks. IEEE Transactions on Wireless Communica- tions 15: 7985–7995.
8. Chatterjee, A., Venkateswaran, P. and Mukherjee, D. (2017) A unified approach of simul- taneous state estimation and anomalous node detection in distributed wireless sensor networks. International Journal of Communication Systems 30: e3191.
9. Chatterjee, A. and Das, K.(2018) State Estimation and Anomaly Detection in Wireless Sensor Networks. In Emerging Wireless Communication and Network Technologies Springer, Singapore, 317–334.
10. Chatterjee, A., Biswas, J. and Das, K. (2020) An automated patient monitoring using discrete-time wireless sensor networks. International Journal of Communication Sys- tems 33: e4390.
11. Daniels, M.J. and Hogan, J.W. (2008) Missing data in longitudinal studies: Strategies for Bayesian modeling and sensitivity analysis. CRC press.
12. Gong, Y., Ruan, F., Fan, Z., Hou, J., Mei, P. and Li, T. (2016) Security Architecture and Requirements for Wireless Sensor Networks. International Journal of Security and Its Applications 10: 295–302.
13. Nagarajan, M., and Karthikeyan, S. (2009) Health Condition Observation with High Level Efficiency in Wireless Sensor Network. Wireless Communication 1: 183–189.
14. Shi-yong, J., Fei, W., Jing-fu, B. and Ting, Y. (2012) Geological disaster monitoring system based on WSN and GSM dual-network integration technology. IEEE 14th International Conference on Communication Technology: 374-379.
15. Winkler, M., Street, M., Tuchs, K.D. and Wrona, K. (2012) Wireless sensor networks for military purposes. In Autonomous Sensor Networks, Springer, Berlin, Heidelberg: 365–394.