Vignette for BayesMSN

Carter Allen

5/12/2020

Installation

To install the `BayesMSN` package, use the code below.

```
devtools::install_github("carter-allen/BayesMSN")
```

Once the package has been installed, you may load it as usual.

```
library(BayesMSN)
```

Overveiw

`BayesMSN` is an `R` package for fitting Bayesian multivariate skew-normal mixture models to longitudinal/repeated measures data that may possibly feature latent sub-clusters of longitudinal outcomes. The model is presented in full detail in Allen et al. (2020). Some key features of this model are as follows:

- Uses finite mixture modeling to explain heterogeneity among repeated measures in terms of a parsimonious set of mixture components (i.e., clusters). This feature of the model is useful if interest lies in uncovering latent clusters in longitudinal outcomes that may not be apparent from marginal trajectories.
- Accounts for the possibility of skewness or heavy tails in repeated measures by utilizing skew-normal or skew-t distributions instead of traditional Gaussian distributions.
- Allows for a wide range of possible covariance patterns among the repeated measures by utlizing unstructured covariance matrices for the multivariate skew-normal (MSN) and multivariate skew-t (MST) distributions.
- Models the cluster indicators themselves using P'olya–Gamma data augmentation to explain cluster membership as a function of covariates of interest. This is a key advantage over other clustering methods that allows for deeper understanding of cluster profiles and readily available practical interpretations of clusters.
- Efficiently imputes intermittent missing repeated measures without globar missing at random assumptions (MAR) through the use of the cluster indicators as a discrete shared parameter to account for unobserved association between the missing mechanism and the missing values themselves.

In summary, the `BayesMSN` package is useful for the longitudinal or multivariate data analysts who seeks a flexible model for uncovering latent sub-clusters among the responses and the ability to explain cluster membership in terms of other practically relevant data. This vignette presents three data analysis examples that showcase some of the key features of `BayesMSN`.

Note: The core functions in `BayesMSN` use Gibb's sampling to obtain samples from the posterior distributions of all model parameters. Like all Bayesian MCMC methods, care must be taken to assess convergence of the MCMC sampler. To perform these diagnostics, we suggest external packages such as `coda`, `bayesplot`, and `label.switching`.

Example 1: Clustered, Skew-Normal Repeated Measures without Missing Data

Included in the `BayesMSN` package is a simulated data set `example1_data`. This data set is meant to simulate repeated measures infant motor development scores as described in Allen et al. (2020). The simulated infant motor development outcomes comprise $J = 4$ repeated measures recorded for $n = 1000$ infants. The total sample is composed of three latent clusters, each with cluster-specific model parameters. We also simulated one continuous time-invariant covariate that can be used to model the development scores, and a separate covariate that can be used in the clustering model. The `example1_data` object is a list of three matrices, `Y`, `X`, and `W`, which contain the repeated measures outcomes, regression covariates, and clustering covariates, respectively, as shown below.

```
str(example1_data)
```

```
## List of 3
##  $ Y: num [1:1000, 1:4] 107 108 109 109 109 ...
##  $ X: num [1:1000, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
##  $ W: num [1:1000, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "" "w1"
```

The most basic model that can be fit with the `BayesMSN` package is a MSN mixture model where all repeated measure outcomes are non-missing, as is the case with `example1_data$Y`, which contains no missing values. To fit such a model, use the `fit_msn()` function as shown below. Note that the MCMC sampler runs at a rate of roughly 100 iterations per minute for this data set on most basic CPUs. Please note that 100 MCMC iterations are used below purely for illustrative purposes, and 100 iterations is almost surely not enough to achieve convergence in most applied settings. For most applications of the method presented in Allen et al. (2020) we use 10000 MCMC iterations with a burn in of 1000.

The `fit_msn()` functions accepts data for each model component as a separate matric. Specifically, `Y` is an $n \times J$ matrix of repeated measures, where the J responses for subject i are contained in the i^{th} row of `Y`. Note that if subjects feature missing responses, the `fit_MSN_impute()` function may be used as described in the next section. The `X` argument is an $n \times p$ matrix of covariates, where the p covariates for subject i are contained in the i^{th} row of `X`. Finally, the matrix `W` is an $n \times r$ matrix containing covariates to be used in the multinomial logit model of cluster indicators. These covariates may be the same as those used in `X`, or they may be separate variables.

```
set.seed(1801)
fit3 <- fit_msn(Y = example1_data$Y,
                X = example1_data$X,
                W = example1_data$W,
                K = 3,
                nsim = 100,
                burn = 0)
```

In the model `fit3` above, the number of clusters $K = 3$ was assumed to be known. However, in most applied settings the true value of K is not known *a priori*. To choose the optimal value for K, we can use WAIC, a goodness of fit measure proposed by Watanabe (2010). The `BayesMSN` package includes a function `waic()`, which computes WAIC for models fit with `fit_msn()`, as demonstrated below.

```
fit3_waic <- waic(fit3)
```

```
fit3_waic
```

```
## [1] 16955.62
```

We can compare the value of WAIC for the three cluster model to those of models fit with other values of K,

say $K = 2$ and $K = 4$, keeping in mind that the goal of modeling infant development data is to explain the the heterogeneity amoung development trajectories with a parsimonious set of development clusters. We note that mispecification of K may lead to MCMC convergence issues.

Once the optimal value for K is chosen, we can inspect the posterior distributions of model parameters returned by the `fit_msn()` function. For full details on the parameters involved in the MSN mixture model see Allen et al. (2020). For cluster specific parameters such as $\mathbf{B}_k$ – the matrix of MSN regression coefficients, a list of length K is returned, where element k of the list refers to the estimated parameters for cluster k. The `fit_msn()` function returns a total of `nsim - burn` posterior estimates of each parameter, as such, the matrices of posterior estimates `nsim - burn` rows, each corresponding to a draw from the posterior distribution of that parameter. For matrix parameters such as $\mathbf{B}_k$, each posterior sample is vectorized to be stored in the matrix of posterior samples returned by `fit_msn()`.

To obtain posterior estimates of the regression parameters in `fit3`, we can utilize the `col_summarize()` function included in the `BayesMSN` package. This function accepts a numeric matrix as input and returns a posterior median and credible interval for each column of data in the matrix. A 95% credible interval is used by default, though this can be controled by changing the `level` argument.

The posterior medians and 95% credible intervals for the MSN regression coefficients for cluster 1 $(\beta_{111}, \beta_{112}, ..., \beta_{141}, \beta_{142})$ are computed below. To see the true values of the parameters used to generate `example1_data`, see Simulation 1 of Allen et al. (2020).

```
col_summarize(fit3$BETA[[1]])
```

```
## [1] "107.33 (94.96, 108.69)" "1.09 (-0.78, 5.38)"     "113.19 (93.6, 114.51)"
## [4] "1.62 (0.13, 8.4)"       "119.2 (91.5, 121.01)"   "2.07 (-0.78, 11.4)"
## [7] "125.19 (90.55, 127.31)" "2.58 (-0.13, 14.19)"
```

These are the posterior estimates of the matrix $\mathbf{B}_1$. In general, the matrix $\mathbf{B}_k$ is constructed as shown below.

$$\mathbf{B}_k = \begin{pmatrix} \beta_{k11} & \cdots & \beta_{k1J} \\ \vdots & \ddots & \vdots \\ \beta_{kp1} & \cdots & \beta_{kpJ} \end{pmatrix}$$

The coefficient in the i^{th} row and j^{th} column of $\mathbf{B}_k$ corresponds to the effect of covariate i on the repeated measures outcome at timepoint j. Each draw from the posterior distribution of $\mathbf{B}_k$ is vectorized and stored in the `BETA[[k]]` matrix returned by the `fit_msn()` function. Thus, finding the medians and credible intervals of each column in `BETA[[k]]` summarizes the post-burn-in posterior distribution of $\mathbf{B}_k$.

Similarly, we can compute the posterior estimates of the parameters for the multinomial regression clustering model as follows.

```
col_summarize(fit3$DELTA)
```

```
## [1] "-0.09 (-0.32, 1.3)" "0.12 (-0.26, 0.45)" "0.19 (-0.21, 0.8)"
## [4] "0.09 (-0.24, 0.42)"
```

As described in Allen et al. (2020), the parameters $\boldsymbol{\delta}_k = (\delta_{k1}, ..., \delta_{kr})$ control the effect of multinomial logit covariate r on each subject's probability of belonging to cluster k. We set $\boldsymbol{\delta}_K = \mathbf{0}_{r\times 1}$ for identifiability. Thus, each row in `DELTA` contains a draw from the posterior distribution of the parameters $\delta_{11}, \delta_{12}, ..., \delta_{1r}, ..., \delta_{K-1,1}, ..., \delta_{K-1,r}$.

Example 2: Clustered, Skew-Normal Repeated Measures with Missing Data

A key feature of the Bayesian MSN mixture model proposed in Allen et al. (2020) is the ability to impute intermittently missing repeated measures under the assumption of conditional ignorability. Specifically, the

model assumes that the missing data mechanism is conditionally ignorable given the cluster indicators z_i. As such, the cluster indicators z_i act as a discrete shared parameter that induces unobserved association between the missing data mechanism and the missing values themselves.

To acheive this, we adopt a model logistic regression model for each R_{ij} – the binary indicator equal to 1 if subject i is missing repeated measure j and zero otherwise. This model includes a random subject specific intercept to induce association among repeated measures taken on the same subject.

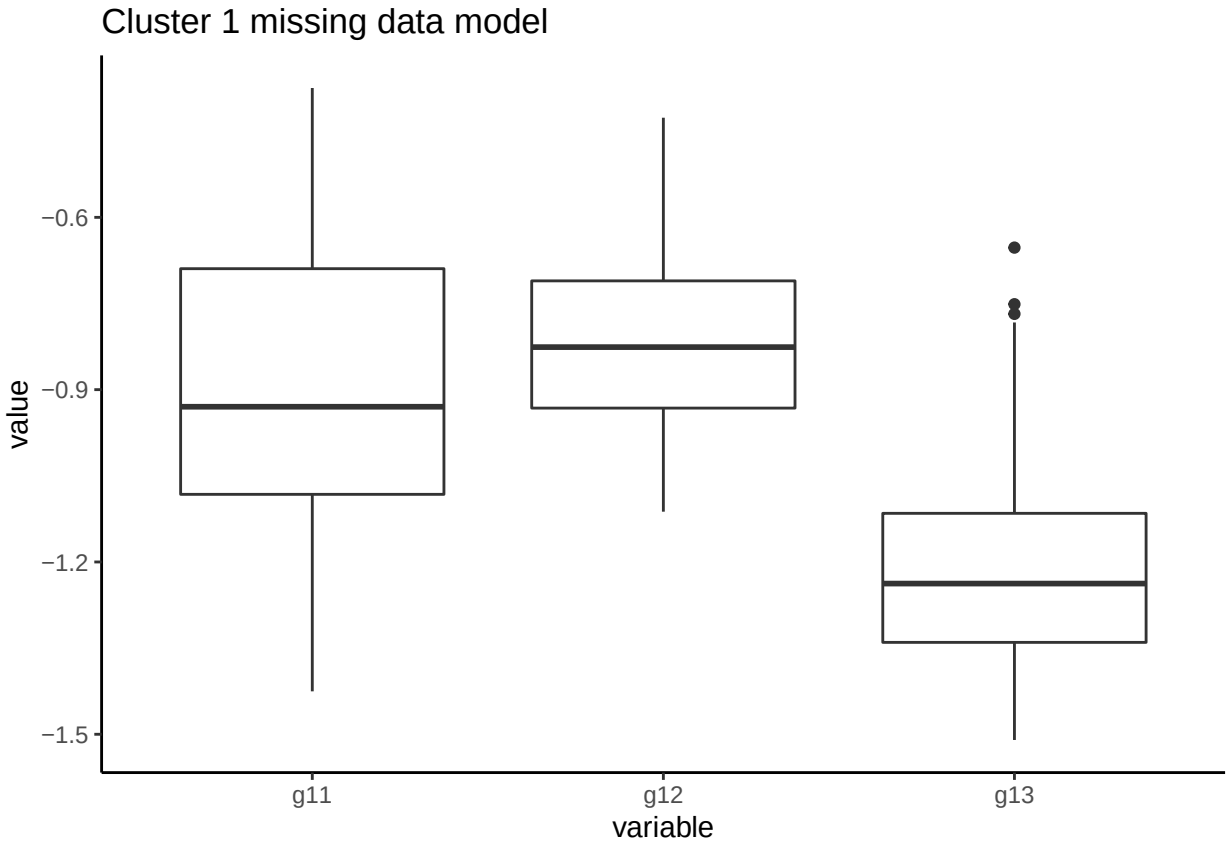
```
fit3_mnar <- fit_msn_impute(example2_data$Y,
                      example2_data$X,
                      example2_data$W,
                      Q = example2_data$Q,
                      K = 3,
                      nsim = 100)
```

Compared to `fit3`, the `fit3_mnar` model includes estimates of the fixed effects parameters used in the random intercept logistic regression model detailed in Section 3 of Allen et al. (2020). These parameters are returned as a list of length K in the `fit3_mnar$GAMMA` object.

```
G1 <-  fit3_mnar$GAMMA[[1]]
G1 <- as.data.frame(G1)
colnames(G1) <- c("g11", "g12", "g13")
```

We can assess the posterior distributions of these parameters by plotting the returned MCMC draws with `ggplot2` as shown below.

```
G1 %>%
  pivot_longer(cols = everything(),
               names_to = "variable",
               values_to = "value") %>%
  ggplot(.,aes(x = variable, y = value)) +
  geom_boxplot() +
  ggtitle("Cluster 1 missing data model") +
  theme_classic()
```



We can see that higher values of these covariates is estimated to have a negative effect on the log-odds of a missing repeated measures response.

Example 3: Clustered, Skew-t Repeated Measures without Missing Data

Finally, as discussed in Allen et al. (2020), we extend the Bayesian MSN mixture model to the Bayesian MST (multivariate skew-t) model to account for the possibility of heavy tails compared to MSN data. The `fit_mst()` function is included in `BayesMSN` to fit the MST model. This function opperates similarly to `fit_msn()`, with the addition of an additional `nu` parameter to specify the degrees of freedom for the MST distribution. Lower values of `nu` correspond to heavier tails relative to the MSN distribution. Consequently, when fit to the `example1_data`, which was generated from an MSN model, we expect the MST model to have poorer fit for lower degrees of freedom relative to the MSN model.

```
fit3_MST_3df <- fit_mst(Y = example1_data$Y,
                        X = example1_data$X,
                        W = example1_data$W,
                        K = 3,
                        nu = 3,
                        nsim = 100,
                        burn = 0)
```

```
fit3_MST_3df_waic <- waic(fit3_MST_3df)
```

```
fit3_MST_3df_waic
```

```
## [1] 19006.87
```

Conclusion

The `BayesMSN` package implements Bayesian multivariate skew normal finite mixture models for clustered longitudinal/multivariate outcomes with potential intermittent missingness. The model allows for skewness and/or heavy tails in repeated measures responses. Clusters are inferred via a multinomial logit regression model of the cluster assignment indicators. Further, the assumtion of conditional ignorability of missing responses relaxed the stricter MAR assumption and allows for modeling of the missing data patterns within clusters using covariates of interest. P'olya–Gamma data augmentation allows for convenient Gibbs updates in the cluster assignment and missing data models. The `BayesMSN` package allows users to conveniently implement the MCMC sampling procedures described in Allen et al. (2020) to quickly obtain posterior samples without needing to implement the MCMC sampler from scratch.

References

Allen, C., Neelon, B., Benjamin-Neelon, S.E. (2020). A Bayesian multivariate mixture model for skewed longitudinal data with intermittent missing observations: An application to infant motor development.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. Journal of Machine Learning Research 11, 3571–3594.