Posterior Predictive Forecasting and Model Assessment

Vineetha Warriyar K. V., Waleed Almutiry and Rob Deardon

March 2020

1 Posterior Predictive Forecasting

Once we have fitted an ILM to the epidemic data (see [2]), we can use it to make forecasts related to future epidemic activity. This can be especially useful in the midst of an unfolding epidemic, allowing us to make predictions about how an epidemic is about to unfold by simulating from a model fitted to the data so far observed. We can do this within the context of EpiILM.

In EpiILM package, we define a function `pred.epi` for getting realizations from the posterior predictive distribution of various epidemiological statistics. Thus, `pred.epi` can be used for forecasting future epidemic activity among the population. The output of the function `pred.epi` is set as an object of class "pred.epi" and it contains a list of objects, some of which are "criterion"' for the used statistic, "crit.sim"' and "crit.obs" for the statistical results of the simulated and observed epidemic data. For graphical illustration of the predictions we introduce another S3 method plot function, `plot.pred.epi`.

To illustrate this, let's consider modelling the spread of a disease through a series of farms ($n = 100$) in a region. We assume that we have no individual-level covariates about the animals or farms, other than whether the farms share the same feed supply feed truck. Now we can generate such an undirected contact network as

```
> set.seed(101)
> n <- 100
> contact <- matrix(0, n, n)
> for(i in 1:(n-1)) {
     contact[i, ((i+1):n)] <- rbinom((n-i), 1, 0.05)
     contact[((i+1):n), i] <- contact[i, ((i+1):n)]
  }
```

The ILM model for this simulated network data would be

$$\mathrm{P(i,t)} = 1 - \exp\{-\alpha \sum_{\mathrm{j \in I(t)}} \mathrm{C_{ij}}\},\ \mathrm{t} = 1, \ldots, \mathrm{t_{max}} \tag{1}$$

Setting $t_{max} = 25$ and $\alpha = 0.1$, we can simulate the epidemic from an SI framework via

```
> set.seed(101)
> netdat <- epidata( type = "SI", n = 100, tmax = 25,  sus.par = 0.1,
                     contact = contact)
```

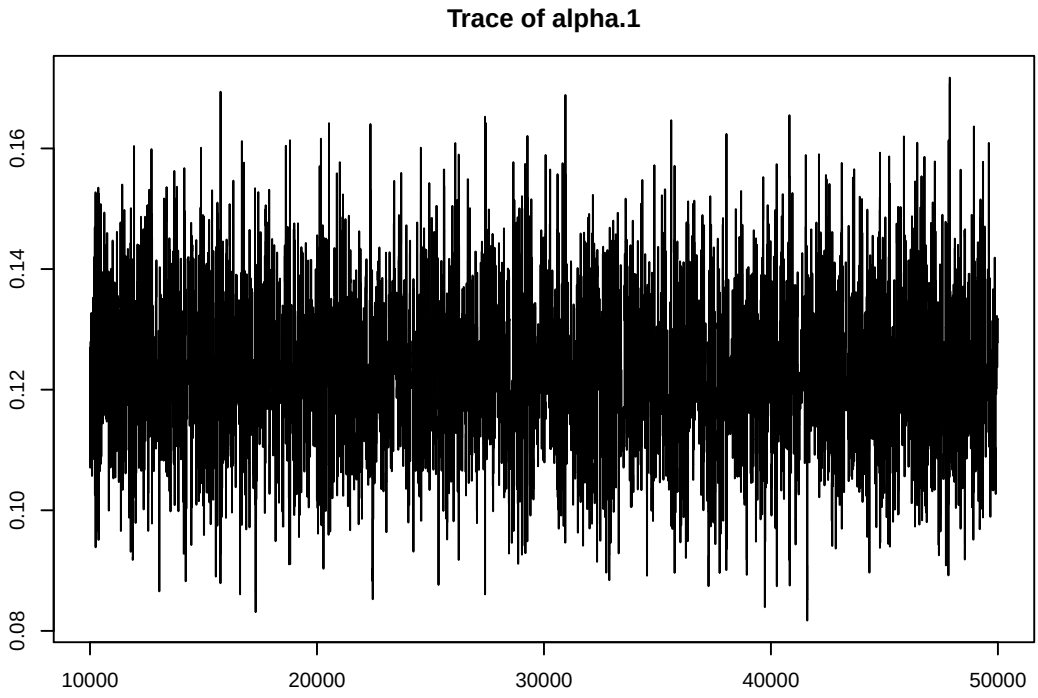
Now let's estimate the unknown parameter α, assuming we know the infection times and contact network. The R code as follows:

```
> t_end <- max(netdat$inftime)
> mcmcout_net <- epimcmc(object = netdat,  tmax = t_end,
          niter = 50000, sus.par.ini = 0.01,pro.sus.var = 0.01,
          prior.sus.dist = "uniform", prior.sus.par = c(0, 10000))
```

generate 50000 samples

The MCMC traceplot for the estimation of the model (1) parameters is given by

```
> plot(mcmcout_net, partype = "parameter", start = 10001, density = FALSE)
```



Having fitted our network-based ILM to the epidemic (`netdat`) in a Bayesian framework, we assume that the epidemic is observed up to time point t. Thus, we can simulate forward from time point t, conditioning upon the infection pattern already observed in the data. For example, assume we have observed the epidemic until time point 15. i.e., $t = 15$. Using `pred.epi` we can simulate, say, 500 posterior predictive forecasts from $t = 15$ by setting `criterion = "newly infectious"`. Following code illustrate the predictions for $t = 15$ and $t = 20$.

```
> set.seed(1001)
> pred.net.15 <- pred.epi(netdat, xx = mcmcout_net, tmin = 15,
                          burnin = 1000, criterion = "newly infectious",
                          n.samples = 500)
```

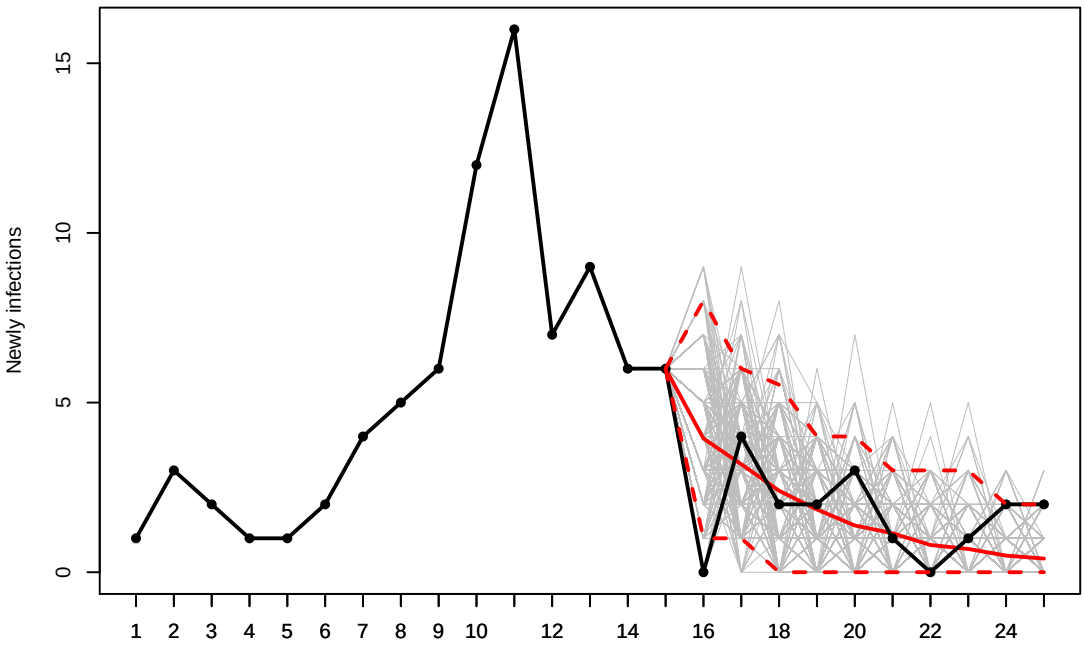
generate 500 epidemics

```
> pred.net.20 <- pred.epi(netdat, xx = mcmcout_net, tmin = 20,
                          burnin = 1000, criterion = "newly infectious",
                          n.samples = 500)
```

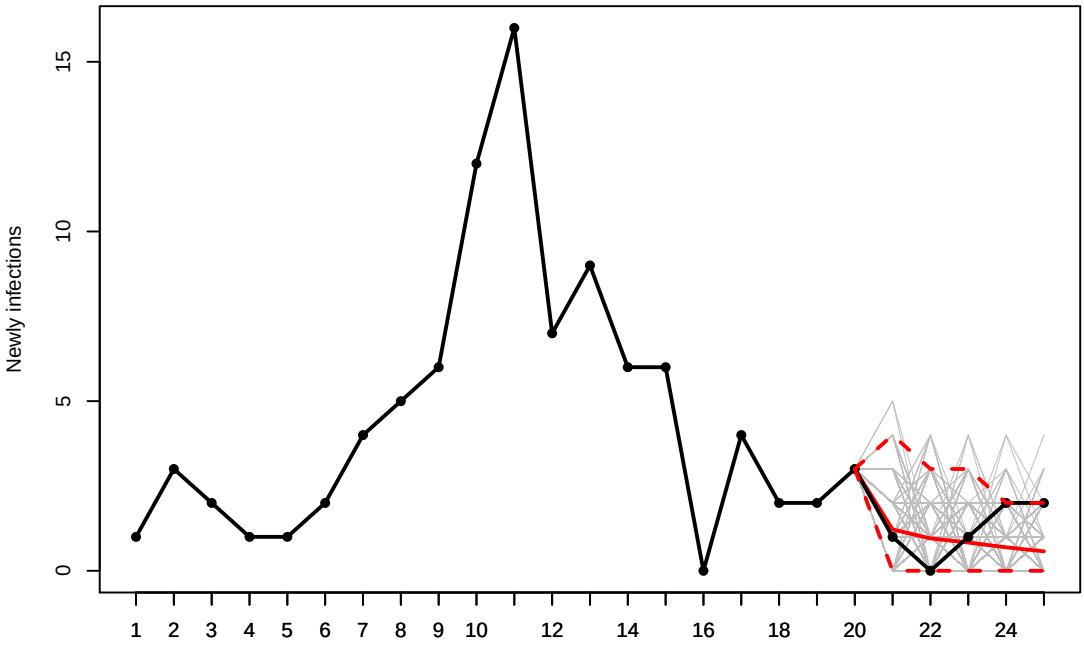
generate 500 epidemics

To plot the above predicted epidemics from time points 15 and 20 we can use the following code. The observed epidemic curve is presented in solid black colour with average of posterior prediction in solid red and corresponding 95% credible intervals (dotted red) for each starting time point.

```
> plot(pred.net.15, col = "red", lwd = 2, pch = 19)
```




```
> plot(pred.net.20, col = "red", lwd = 2, pch = 19)
```



2 Model Selection

Another important feature in this package is the techniques that can be used for model assessment. In general, of course, we are not analyzing simulated data, but real data for which we do not know the underlying model. Once we have the posterior distribution of our statistic of interest, we can compare it with observed data either via a posterior predictive p value or simply graphically. In infectious disease models, we commonly use some form of epidemic curve as our (multivariate) statistical interest, e.g., the number of newly infectious individuals over time. Other commonly used statistics are the length of the epidemic or the time of, or prevalence at, the peak of the epidemic. Again, using `pred.epi` we can obtain the posterior predictive distributions of these statistics for model assessment and the type of the statistic can be specified through the argument `criterion` with three options: "newly infectious" for the number of newly infectious individuals over time, "epidemic length" for the length of the epidemic, and "peak time" for the time of, or prevalence at, the peak of the epidemic. We can also calculate the deviance information criterion (DIC) to compare the fit of different models ([1]) using the `epidic` function. The input for this function consists of log-likelihood values of the MCMC output and the log-likelihood under the posterior mean estimates of the parameters.

We now show how this can be done for the analysis of a simulated spatial epidemic, using

the true underlying fitted model. Consider a spatial ILM model

$$\mathrm{P(i,t)} = 1 - \exp\{-(\alpha_0 + \alpha_1 \mathrm{A(i)}) \sum_{\mathrm{j}\in \mathrm{I(t)}} \mathrm{d}_{\mathrm{ij}}^{-\beta}\},\ \mathrm{t} = 1, \ldots, \mathrm{t}_{\max}, \tag{2}$$

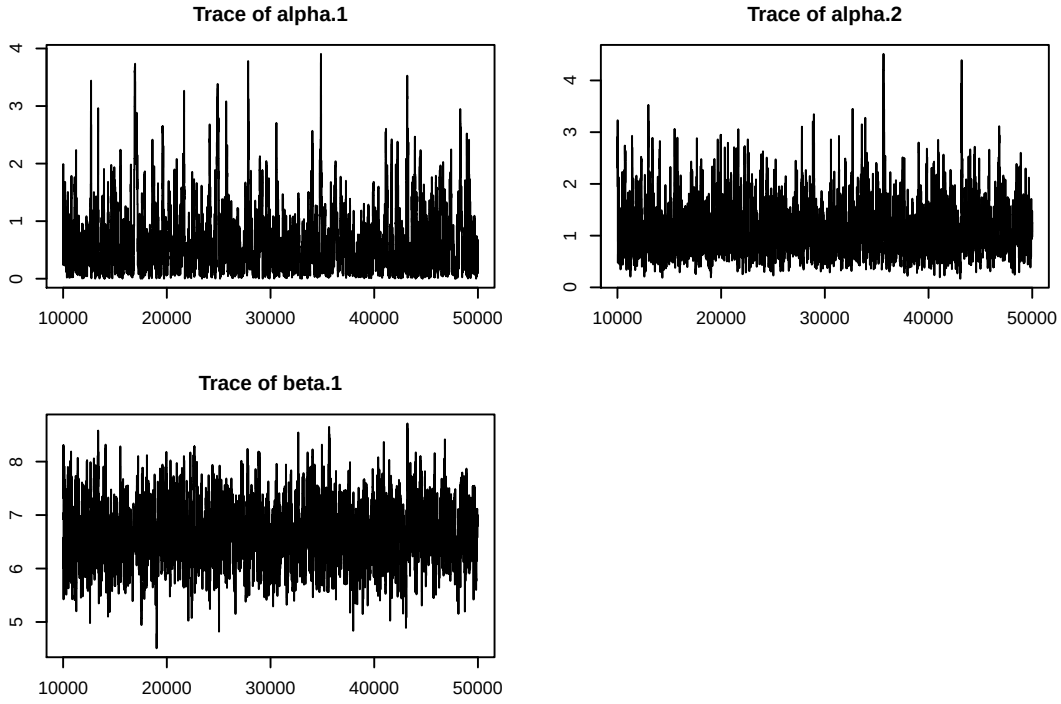
which models the spread of a highly transmissible disease through a series of farms, say $n = 100$. In model 2, assume that the susceptibility covariate A represents the number of animals on each farm, α_0 is the baseline susceptibility, α_1 is the number of animals effect, and β is the spatial parameter. Suppose that the spatial locations of the farms and the number of animals on each farm are known and set the parameters $(\alpha_0, \alpha_1) = (0.5, 0.5)$ and $\beta = 6$. In this situation, it is reasonable to treat the farms themselves as individual units. Also we treat the extent of infection from outside the observed population of farms as unknown ($\varepsilon = 0$) and set $t_{max} = 25$. Considering an SI compartmental framework for this situation, the epidemic is simulated using the following

```
> # simulate spatial locations
> set.seed(101)
> x <- runif(100, 0, 10)
> y <- runif(100, 0, 10)
> # simulate covariate, number of animals on each farm
> A <- round(rexp(100,1/50))
> SI.cov <- epidata(type = "SI", n = 100, tmax = 25, x = x, y = y,
                    Sformula = ~A, sus.par = c(0.5, 0.5), beta = 6)
```

We can now refit the generating model to this simulated data and consider the posterior estimates of the model parameters.

```
> t_end <- max(SI.cov$inftime)
> prior_par <- matrix(rep(1, 4), ncol = 2, nrow = 2)
> mcmcout_M1 <- epimcmc(SI.cov, Sformula = ~A, tmax = t_end, niter = 50000,
                        sus.par.ini = c(0.001, 0.001), beta.ini = 0.01,
                        pro.sus.var = c(0.08, 0.4), pro.beta.var = 0.5,
                        prior.sus.dist = c("gamma","gamma"),
                        prior.sus.par = prior_par,
                        prior.beta.dist = "uniform", prior.beta.par = c(0, 10000) )
  generate 50000 samples
> summary(mcmcout_M1, start = 10001)
Model: SI distance-based discrete-time ILM
Method: Markov chain Monte Carlo (MCMC)

Iterations = 10001:50000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 40000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:
```

```
          Mean     SD Naive SE Time-series SE
alpha.1 0.5965 0.5652 0.002826        0.02782
alpha.2 1.0708 0.5029 0.002515        0.01703
beta.1  6.6063 0.5679 0.002839        0.01820

2. Quantiles for each variable:

          2.5%    25%    50%    75% 97.5%
alpha.1 0.0209 0.1898 0.4339 0.8254 2.099
alpha.2 0.3906 0.7120 0.9732 1.3302 2.275
beta.1  5.5619 6.2136 6.5939 6.9909 7.735
> #  MCMC traceplot for the estimation of the model (2) parameters
> plot(mcmcout_M1, partype = "parameter", start = 10001, density = FALSE)
```



Let's assume we wish to fit a spatial model that does not include the number of animals effect, i.e., a model of the form:

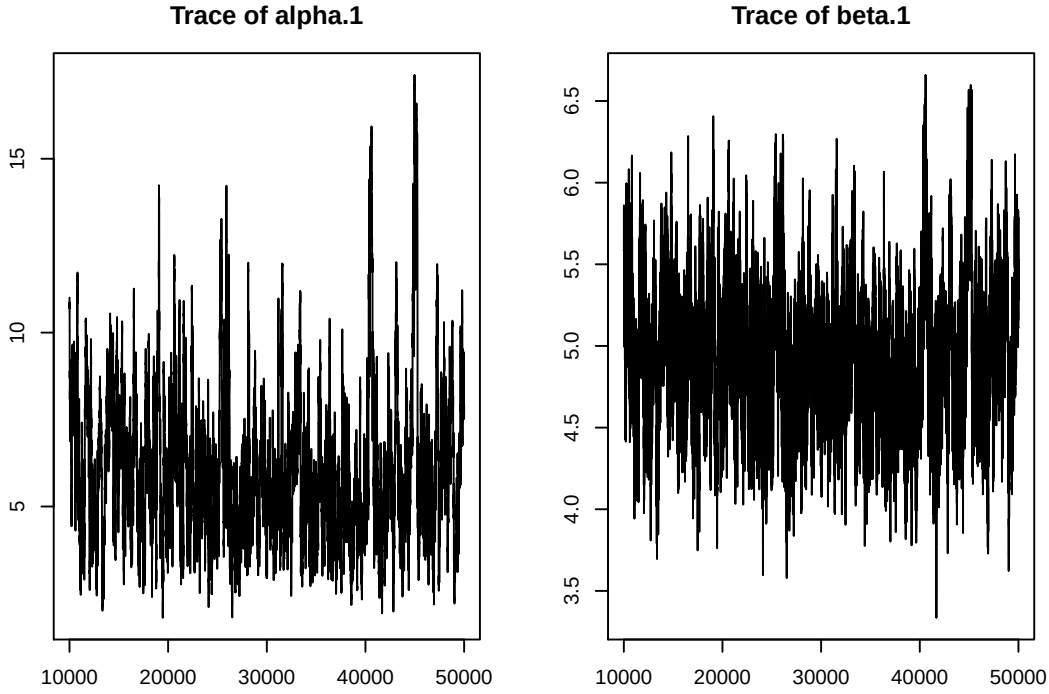
$$\mathrm{P}(\mathrm{i},\mathrm{t}) = 1 - \exp\{-\alpha_0 \sum_{\mathrm{j}\in\mathrm{I}(\mathrm{t})} \mathrm{d}_{\mathrm{ij}}^{-\beta}\},\ \mathrm{t} = 1,\ldots,50 \tag{3}$$

where α_0 is the susceptibility constant and β is the spatial parameter. To estimate the unknown parameters α_0 and β, we could again use the function `epimcmc`

```
> set.seed(101)
> mcmcout_M2 <- epimcmc(SI.cov, tmax = t_end, niter = 50000, sus.par.ini = 0.01,
                        beta.ini = 0.01,  pro.sus.var = 0.1, pro.beta.var = 0.5,
                        prior.sus.dist = "uniform",
                        prior.sus.par = c(0, 10000), prior.beta.dist  = "uniform",
                        prior.beta.par = c(0, 10000))
```

generate 50000 samples The estimate of the posterior mean of the parameters and 95% credible intervals after 10000 iterations of burn-in have been removed are

```
> summary(mcmcout_M2,  start = 10001)
Model: SI distance-based discrete-time ILM
Method: Markov chain Monte Carlo (MCMC)

Iterations = 10001:50000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 40000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

         Mean     SD Naive SE Time-series SE
alpha.1 6.008 2.2015  0.01101        0.16903
beta.1  4.904 0.4301  0.00215        0.02748

2. Quantiles for each variable:

         2.5%   25%   50%   75%  97.5%
alpha.1 2.926 4.418 5.575 7.151 11.362
beta.1  4.139 4.600 4.886 5.177  5.832
> # MCMC traceplot for the estimation of the model (3) parameters
> plot(mcmcout_M2, partype = "parameter", start = 10001, density = FALSE)
```



For model comaprison let's say, we are interested in the epidemic curve in the form of the number of newly infected individuals over time. First, we will generate 500 posterior predictive epidemics using the MCMC output obtained from model (2) and (3)

```
> set.seed(101)
> pred.model1 <- pred.epi(SI.cov, Sformula = ~A, xx = mcmcout_M1,
                          criterion = "newly infectious",  n.samples = 500)
```

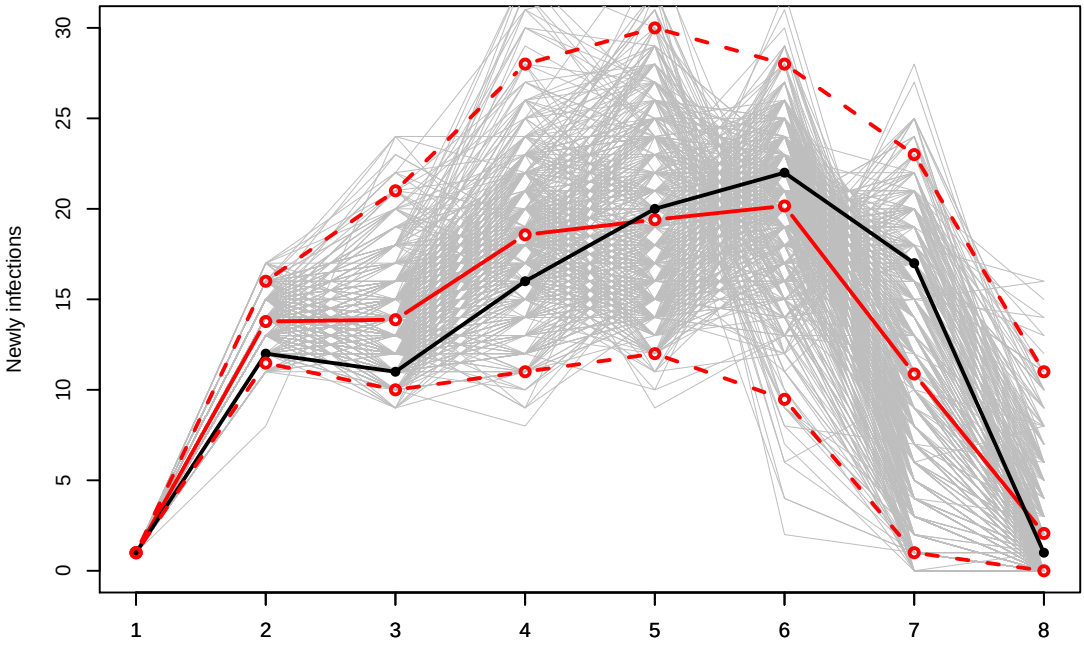
generate 500 epidemics

```
> # convert ILM model (2) into epidata object
> model2.data <- as.epidata(type = "SI", n = 100, x = x, y = y,
                               inftime = SI.cov$inftime)
> pred.model2 <- pred.epi(model2.data, xx = mcmcout_M2,
                           criterion = "newly infectious", n.samples = 500)
```

generate 500 epidemics

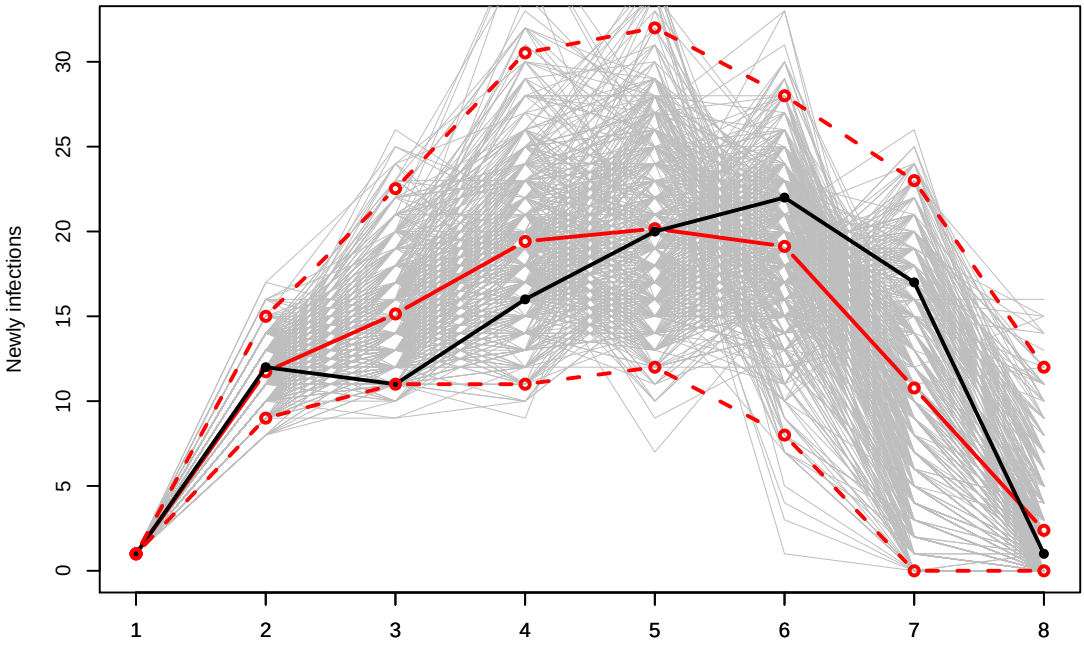
In order to assess the model fit, we can plot the posterior predictive realizations, the time-wise 95% credible intervals, and the true epidemic curve from the above posterior predictions. The observed epidemic curve (solid black) with average of posterior prediction (solid red) and corresponding 95% credible intervals for model (2) and (3) are shown in the following figures. The grey lines are the 500 MCMC samples. For model 2

```
> plot(pred.model1, col = "red", type = "b", lwd = 2)
```



and for model 3

```
> plot(pred.model2, col = "red", type = "b", lwd = 2)
```



We can see from the above posterior predictive plots that, both the correct model and mis-specified model fare similarly well when considering a comparison of the original epidemic curve and the posterior predictive distribution of said curve, although the posterior uncertainty is greater under the mis-specified model.

We can also calculate the deviance information criterion (DIC) to compare the fit of our two models.The following code is used for the model (2):

```
> loglike <- epilike(SI.cov, tmax = t_end, Sformula = ~A, sus.par = c(0.597, 1.071),
                                                  beta = 6.606)
> dic1 <- epidic(burnin = 10000, niter = 50000, LLchain = mcmcout_M1$Loglikelihood,
                                                  LLpostmean = loglike)
> dic1
[1] 134.9902
```

and the DIC for model (3) is

```
> loglike <- epilike(model2.data, tmax = t_end,  sus.par = 6.210, beta = 4.942)
> dic2 <- epidic(burnin = 10000, niter = 50000, LLchain = mcmcout_M2$Loglikelihood,
                                                   LLpostmean = loglike)
```

The noticeably lower DIC value (134.99) for model (2) implies a significantly better fit for the true model (as we would expect).

References

[1] David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64:583–639, 2002.

[2] Vineetha Warriyar K. V., Waleed Almutiry, and Rob Deardon. Individual-level modelling of infectious disease data: Epiilm. Preprint available at `arXiv:2003.04963[stat.AP]`.