Sandia National Laboratories

SAND2020-13331O

PRIME - A Software Toolkit for the Characterization of Partially Observed Epidemics in a Bayesian Framework

Patrick Blonigan, Kamaljit Chowdhary, Jaideep Ray
Cosmin Safta

Nov 29, 2020

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

GETTING STARTED:

CHAPTER

ONE

ABOUT

PRIME is a modeling framework designed for the "real-time'" characterization and forecasting of partially observed epidemics. Characterization is the estimation of infection spread parameters using daily counts of symptomatic patients. The method is designed to help guide medical resource allocation in the early epoch of the outbreak. The estimation problem is posed as one of Bayesian inference and solved using a Markov Chain Monte Carlo technique. The framework can accommodate multiple epidemic waves and can help identify different disease dynamics at the regional, state, and country levels. We include examples using publicly available COVID-19 data.

1.1 Team

PRIME was written and developed by Cosmin Safta, Jaideep Ray, Patrick Blonigan, and Kamaljit Chowdhary. We would like to acknowledge helpful suggestions made by several colleagues: Erin Acquesta, Thomas Catanach, Bert Debusschere, Sean DeRosa, Pat Finley, Edgar Galvan, Gianluca Geraci, John D. Jakeman, Mohammad Khalil, Khachik Sargsyan, and Teresa Portone.

1.2 Acknowledgments

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This work was funded in part by the Laboratory Directed Research & Development (LDRD) program at Sandia National Laboratories and by the US Department of Energy (DOE) Office of Science through the National Virtual Biotechnology Laboratory, a consortium of national laboratories (Argonne, Los Alamos, Oak Ridge, and Sandia) focused on responding to COVID-19, with funding provided by the Coronavirus CARES Act. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

1.3 Requirements

The following python packages are required for PRIME, in addition to other default python packages.

- dateutil, h5py, matplotlib, numpy, scipy

We have tested PRIME with python versions 3.6-3.8.

CHAPTER

TWO

MODEL DESCRIPTION

PRIME implements an epidemiological model to characterize and forecast the rate at which people turn symptomatic from disease over time. For the purpose of this model, we assume that once people develop symptoms, they have ready access to medical services and can be diagnosed readily. From this perspective, these forecasts represent a lower bound on the actual number of people that are infected with COVID-19 as the people currently infected, but still incubating, are not accounted for. A fraction of the population infected might also exhibit minor or no symptoms at all and might not seek medical advice. Therefore, these cases will not be part of patient counts released by health officials. The epidemiological model consists of two canonical elements: an infection rate model and an incubation rate model. One or more infection rate models are then combined through a convolution with the incubation rate model to yield the number of cases that turn symptomatic daily.

2.1 Infection Rate Model

The infection rate component is modeled as a Gamma distribution

$$f_\Gamma(t; k_j, \theta_j) = \theta_j^{-k_j} t^{k_j - 1} \exp(-t/\theta_j)/\Gamma(k_j) \tag{2.1}$$

with shape k_j and scale θ_j. The choice of values for the pair (k_j,lthjl) can accomodate both sharp increases in the number of infections, which would correspond to strained medical resources, as well as weaker gradients corresponding to a smaller pressure on the available medical resources. Fig. 2.1 show example infection rate curves for several shape and scale parameter values.

2.2 Incubation Rate Model

PRIME employs a lognormal incubation distribution for COVID-19 [Lauer2020]. The probability density function (PDF), f_{LN}, and cumulative distribution function (CDF), F_{LN}, of the lognormal distribution are given by

$$f_{LN}(t; \mu, \sigma) = \frac{1}{t\sigma\sqrt{2}} \exp\left(-\frac{(\log t - \mu)^2}{2\sigma^2}\right), \; F_{LN}(t; \mu, \sigma) = \frac{1}{2}\mathrm{erfc}\left(-\frac{\log t - \mu}{\sigma\sqrt{2}}\right) \tag{2.2}$$

In this toolkit we model the mean μ as Student's t distribution with $n = 36$ degrees of freedom which provided the closest agreement for the 95% confidence interval with the data in [Lauer2020]. Similarly, the standard deviation σ is assumed to have a chi-square distribution. The resulting 95% CIs are $[1.48, 1.76]$ and $[0.320, 0.515]$ for μ and σ, respectively. The left frame in Fig. 2.2 shows the family of PDFs with μ and σ drawn from Student's t and χ^2 distributions, respectively. The nominal incubation PDF is shown in black in this frame. The impact of the uncertainty in the incubation model parameters is displayed in the right frame of this figure. For example, 7 days after infection, there is a large variability (60%-90%) in the fraction of infected people that completed the incubation phase and started displaying symptoms. This variability decreases at later times, e.g. after 10 days more then 85% of case completed the incubation process.

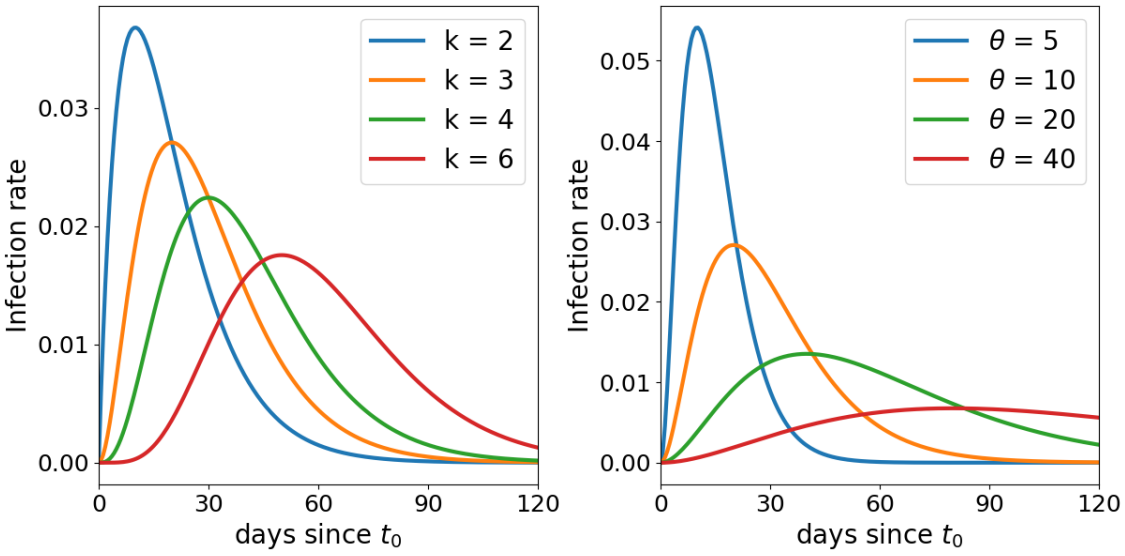

Fig. 2.1: Infection rate models with fixed scale parameters $\theta = 10$ (left frame) and fixed shape parameter $k = 3$ (right frame).

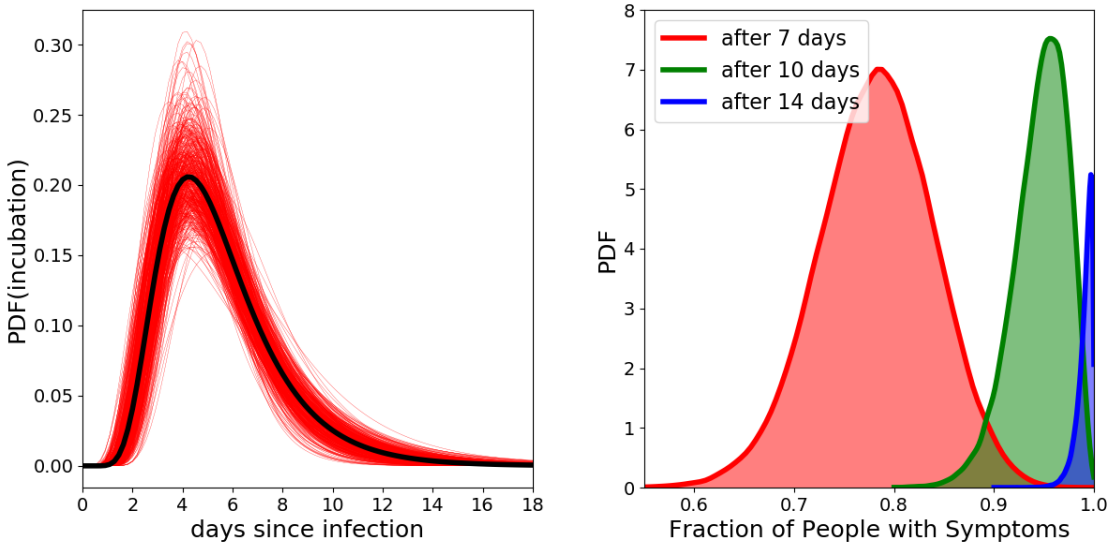

Fig. 2.2: Probability density functions for the incubation model (left frame) and fraction of people for which incubation ran its course after 7, 10, and 14 days respectively (right frame)

2.3 Single Wave Model

With these assumptions the number of people infected *and* with completed incubation period in the time range $[t_{i-1}, t_i]$ can be written as a convolution between the infection rate and the incubation rate [Safta2020]

$$n_i \approx N(t_i - t_{i-1}) \int_{t_0}^{t_i} f_\Gamma(\tau - t_0; k, \theta) f_{LN}(t_i - \tau; \mu, \sigma) d\tau \tag{2.3}$$

Here N represents the total number of cases over the course of the epidemic, $(t_i - t_{i-1})$ is typically equal to 1 day, and the time parameter t_0 represents the start of the epidemic. In the expression above the sub-script *j* was neglected since the model contains one wave only.

2.4 Multi-Wave Model

The multiple wave model is an extension of the single wave model presented above. In the multi-wave model, a set of infection curves are superimposed to approximate the evolution of the epidemic that exhibits multiple peaks across certain regions. The resulting model is written as

$$n_i \approx (t_i - t_{i-1}) \int_{t_0}^{t_i} \left(\sum_{j=1}^{K} N_j f_\Gamma(\tau - t_0 - \Delta t_j; k_j, \theta_j) \right) f_{LN}(t_i - \tau; \mu, \sigma) d\tau \tag{2.4}$$

Here N_j represents the total number of cases over the course of the *j-th* wave, and Δt_j represents the time shift for the *j-th* infection curve with respect to the start of the epidemic t_0.

CHAPTER

THREE

DATA

The number of people developing symptoms daily are compared to data obtained from several sources at the national, state, or regional levels [JHUCOVID19] [NYTCOVID19]. We found that, for some states or regions, the reported daily counts exhibited a significant amount of noise. This is caused by variation in testing capabilities and sometimes by how data is aggregated from region to region over the coarse of a week. To filter the noise observed in daily case count data, we use of 7-day rolling averages. Time series of daily counts, raw data with black symbols and filtered with red symbols, are presented in Fig. 3.1. We employ filtered data in the examples shown in this manual.

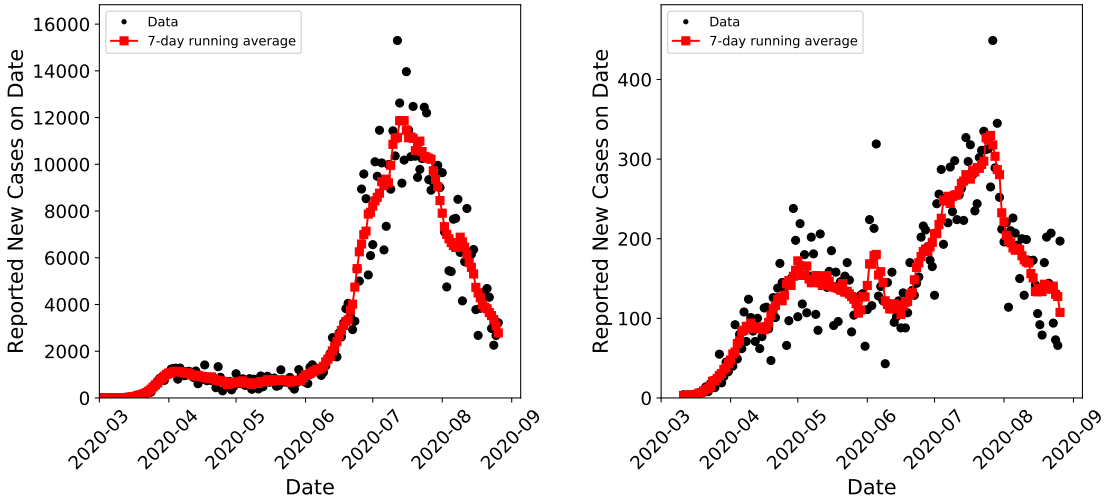

Fig. 3.1: Daily confirmed cases of COVID-19 aggregated at state level, shown in black symbols, and the corresponding 7-day averaged data shown with red lines and symbols.

CHAPTER

FOUR

BAYESIAN FRAMEWORK

Given data, $\boldsymbol{y}$, in the form of time-series of daily counts and the model predictions $\boldsymbol{n}$ for the number of new symptomatic counts daily for the same time range, we employ a Bayesian framework to calibrate the epidemiological model parameters. The discrepancy between the data and the model is written as

$$\boldsymbol{y} = \boldsymbol{n}(\Theta) + \epsilon \tag{4.1}$$

Here, Θ is a vector of model parameters, and ϵ represents the statistical discrepancy between the model and the data. The elements of Θ depend on the number of epidemic waves being modeled.

$$\Theta = \Theta^{(1)} \cup \Theta^{(2)} \cup \ldots \cup \Theta^{(K)} \cup \Theta^{(\epsilon)} \tag{4.2}$$

where $\Theta^{(j)}$ are the parameter for the *j*-th wave of infections, *K* is the number of waves and $\Theta^{(\epsilon)}$ are parameters for the error model. The parameters for each wave are given by

$$\Theta^{(j)} = \{\Delta t_j, N_j, k_j, \theta_j\} \tag{4.3}$$

For the first epidemic wave $\Delta t_1 = t_0$.

The error model encapsulates, in this context, both errors in the observations as well as errors due to imperfect modeling choices. The observation errors include variations due to testing capabilities as well as errors when tests are interpreted. Values for the vector of parameters Θ is estimated in the form of a multivariate PDF via Bayes theorem

$$p(\Theta|\boldsymbol{y}) \propto p(\boldsymbol{y}|\Theta)p(\Theta) \tag{4.4}$$

where $p(\Theta|\boldsymbol{y})$ is the posterior distribution we are seeking after observing the data $\boldsymbol{y}$, $p(\boldsymbol{y}|\Theta)$ is the likelihood of observing the data $\boldsymbol{y}$ for a particular set of values for the model parameters Θ, and $p(\Theta)$ encapsulates any prior information available for the model parameters. Bayesian methods are well-suited for dealing with heterogeneous sources of uncertainty, in this case from our modeling assumptions, i.e. model and parametric uncertainties, as well as the communicated daily counts of COVID-19 new cases, i.e. experimental errors.

4.1 Likelihood Construction

The library provides options for both deterministic and stochastic formulations for the incubation model. In the former case the mean and standard deviation of the incubation model are fixed at their nominal values and the model prediction n_i for day t_i is a scalar value that depends on $Theta$ only. In the latter case, the incubation model is stochastic with mean and standard deviation of its natural logarithm treated as Student's t and chi^2 random variables, respectively, as discussed in *Incubation Rate Model* Section. Let us denote the underlying independent random variables by $\boldsymbol{\xi} = \{\xi_\mu, \xi_\sigma, \}$. The model prediction $n_j(\boldsymbol{\xi})$ for day *j* is now a random variable induced by ${bm xi}$ plugged into *Single Wave Model* or *Multi-Wave Model* and $\boldsymbol{n}(\boldsymbol{\xi})$ is a random vector.

4.1.1 Deterministic Incubation Model

PRIME provides options for both Gaussian and negative binomial formulations for the statistical discrepancy ϵ between $\boldsymbol{n}$ and $\boldsymbol{y}$. In the first approach we assume ϵ has a zero-mean Multivariate Normal (MVN) distribution. Given the sparsity of data, correlations across time are currently neglected and the liklihood $p(\boldsymbol{y}|\Theta)$ is computed as

$$p(\boldsymbol{y}|\Theta) = \prod_{i=1}^{D} \pi_{n_i(\Theta)}(y_i) = (2\pi)^{-D/2} \prod_{i=1}^{D} \sigma_i^{-1} \exp\left(-\frac{(y_i - n_i)^2}{2\sigma_i^2}\right) \tag{4.5}$$

with

$$\sigma_i = \sigma_a + \sigma_m \, n_i(\Theta) \tag{4.6}$$

The additive, σ_a, and multiplicative, σ_a, components of the error model $\Theta^{(\epsilon)} = \{\sigma_a, \sigma_a\}$ will be inferred jointly with the model parameters. In practive, PRIME infers the logarithm of these parameters to ensure they remain positive.

The second approach assumes a negative-binomial distribution for the discrepancy between data and model predictions. The negative-binomial distribution is commonly used in epidemiology to model overly dispersed data, e.g. in case where the standard deviation exceeds the mean [Lloyd2007]. For this modeling choice, the likelihood of observing the data given a choice for the model parameters is given by

$$p(\boldsymbol{y}|\Theta) = \prod_{i=1}^{D} \pi_{n_i(\Theta)}(y_i) = \prod_{i=1}^{D} \binom{y_i + \alpha - 1}{\alpha - 1} \left(\frac{\alpha}{\alpha + n_i(\Theta)}\right)^{\alpha} \left(\frac{n_i(\Theta)}{\alpha + n_i(\Theta)}\right)^{y_i} \tag{4.7}$$

where $\alpha > 0$ is the dispersion parameter, and

$$\binom{y_i + \alpha - 1}{\alpha - 1} = \frac{\Gamma(y_i + \alpha)}{\Gamma(\alpha)\Gamma(y_i + 1)} \tag{4.8}$$

is the binomial coefficient. For simulations employing a negative binomial distribution of discrepancies, the logarithm of the dispersion parameter α will be inferred jointly with the other model parameters.

4.1.2 Stochastic Incubation Model

For the {it stochastic incubation model} the likelihood reads as

$$p(\boldsymbol{y}|\Theta) = \pi_{\boldsymbol{n}(\Theta),\boldsymbol{\xi}}(\boldsymbol{y}) \approx \prod_{i=1}^{D} \pi_{n_i(\Theta),\boldsymbol{\xi}}(y_i) \tag{4.9}$$

The second expression in the right-hand side above assumes independence of the discrepancies between different days. Unlike the deterministic incubation model, the likelihood components for each day $\pi_{n_i(\Theta),\boldsymbol{\xi}}(y_i)$ are not analytically tractable anymore since they now incorporate contributions from $\boldsymbol{\xi}$, i.e. from the variability of the parameters of the incubation model. One can evaluate the likelihood via kernel density estimation (KDE) by sampling $\boldsymbol{\xi}$ for each sample of Θ, and combining these samples with samples of the assumed discrepancy ϵ, in order to arrive at an estimate of $\pi_{n_i(\Theta),\boldsymbol{\xi}}(y_i)$. In fact, by sampling a *single* value of $\boldsymbol{\xi}$ for each sample of Θ, one achieves an unbiased estimate of the likelihood $\pi_{n_i(\Theta),\boldsymbol{\xi}}(y_i)$, and given the independent-component assumption, it also leads to an unbiased estimate of the full likelihood $\pi_{\boldsymbol{n}(\Theta),\boldsymbol{\xi}}(\boldsymbol{y})$.

4.2 Posterior Distribution Sampling

A Markov Chain Monte Carlo (MCMC) algorithm is used to sample from the posterior density $p(\Theta|\boldsymbol{y})$. MCMC is a class of techniques that allows sampling from a posterior distribution by constructing a Markov Chain that has the posterior as its stationary distribution. In particular, PRIME uses a an adaptive Metropolis algorithm [Haario2001]. Given the construction corresponding to the stochastic incubation model presented above, we employ the unbiased estimate of the approximate likelihood. This is the essence of the pseudo-marginal MCMC algorithm [Andrieu2009] guaranteeing that the accepted MCMC samples correspond to the posterior distribution. At each MCMC step we draw a random sample ξ from its distribution, and then we estimate the likelihood in a way similar to the deterministic incubation model.

Fig. 4.1 displays 1D and 2D joint marginal distributions based on two-wave model results. We use the Raftery-Lewis diagnostic [Raftery1992] to determine the number of MCMC samples required for converged statistics corresponding to stationary posterior distributions for Θ. The required number of samples is of the order $o(10^5 - 10^6)$ depending on the geographical region employed in the inference. The resulting Effective Sample Size [Kass1998] varies between *8,000* and *15,000* samples depending on each parameter which is sufficient to estimate joint distributions for the model parameters.

4.3 Posterior Predictive Tests

We will employ Bayesian posterior-predictive distributions [Lynch2004] to assess the predictive skill of the statistical model. The Bayesian posterior-predictive distribution, defined below is computed by marginalization of the likelihood over the posterior distribution of model parameters Θ:

$$p_{\mathrm{pp}}\left(\boldsymbol{y}^{(\mathrm{pp})}|\boldsymbol{y}\right) = \int_{\boldsymbol{\Theta}} p(\boldsymbol{y}^{(\mathrm{pp})}|\Theta)p(\Theta|\boldsymbol{y})d\Theta. \tag{4.10}$$

The posterior predictive distribution $p_{\mathrm{pp}}\left(\boldsymbol{y}^{(\mathrm{pp})}|\boldsymbol{y}\right)$ is estimated through sampling, using the parameter samples readily available from the MCMC exploration of the parameter space, i.e. similar to results shown in Fig.~ref{fig:mcmc}. Typically we subsample the MCMC chain to about 10-15K samples that will be used to generate posterior predictive statistics. After the model evaluations $\boldsymbol{y} = \boldsymbol{n}(\Theta)$ are completed, we add random noise consistent with the likelihood model settings presented in ref{sec:lk}. The resulting samples are used to compute summary statistics corresponding to $p_{\mathrm{pp}}\left(\boldsymbol{y}^{(\mathrm{pp})}|\boldsymbol{y}\right)$.

The posterior-predictive distribution results can be used in hindcast mode, to check how well the model follows the data, and for short-term forecasts for the spread dynamics of this disease. In the hindcast regime, the infection rate is convolved with the incubation rate model to generate statistics for $\boldsymbol{y}^{(\mathrm{pp})}$ that will be compared against $\boldsymbol{y}$, the data used to infer the model parameters. The same functional form can be used to generate statistics for $\boldsymbol{y}^{(\mathrm{pp})}$ beyond the set of dates for which data was available. We limit these forecasts to 7–10 days as our infection rate model does not count for changes in social dynamics that can significantly impact the epidemic over a longer timerange.

4.4 Model Selection

Quantitative comparisons between models can be made with several metrics defined in the following sections.

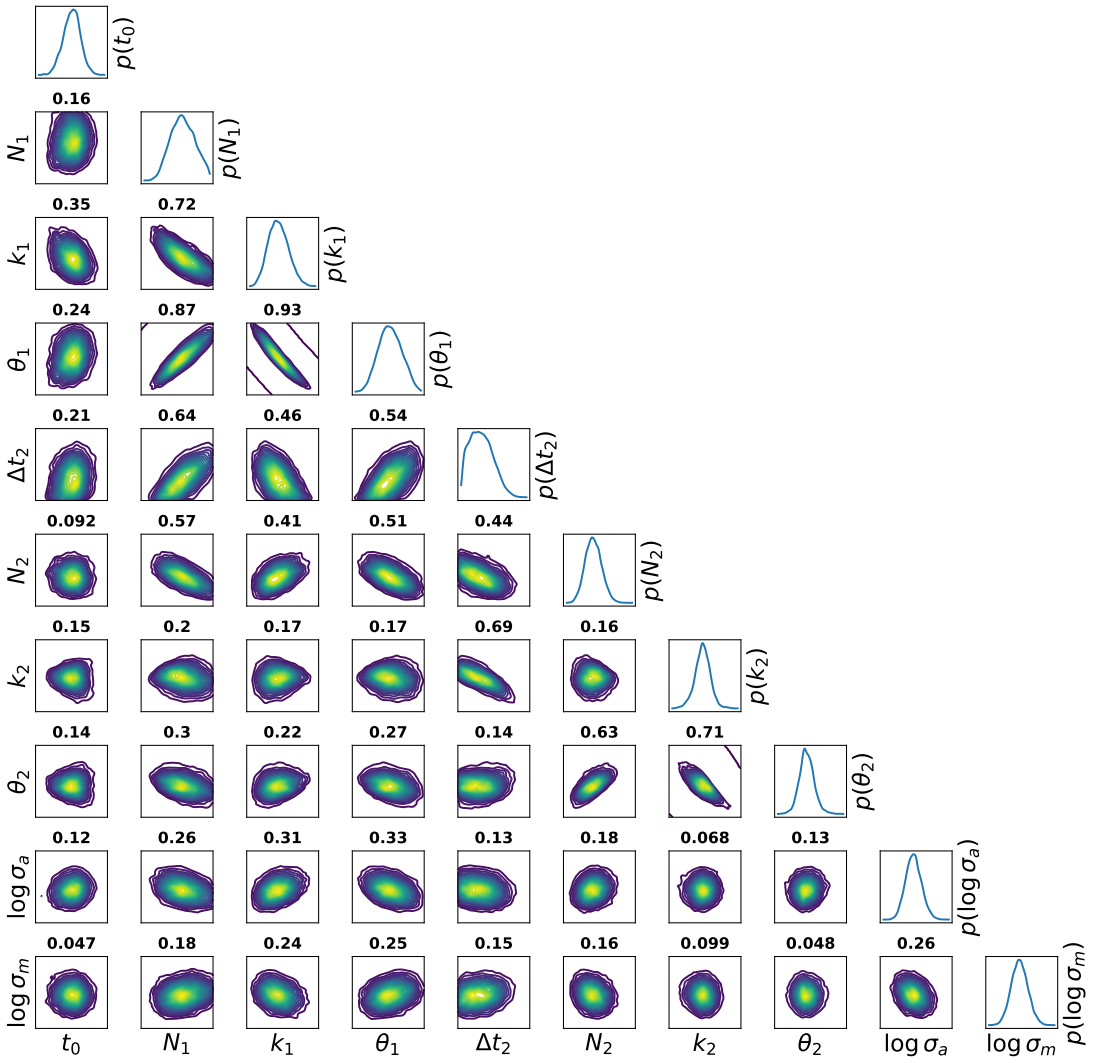

Fig. 4.1: 1D and 2D joint marginal distributions the components of $\Theta = \{t_0, N_1, k_1, \theta_1, \Delta t_2, N_2, k_2, \theta_2, \log \sigma_a, \log \sigma_m\}$ for data from California up to 2020-08-19. Distance correlations for each pair of parameters is displayed above each joint marginal distribution.

4.4.1 AIC

The Akaike Information Criteria (AIC) [Akaike1974] is defined as

$$AIC = 2m_\Theta - 2\ln(L_{max}), \tag{4.11}$$

where m_Θ is the number of parameters in Θ and L_{max} is the maximum value of the likelihood $p(\boldsymbol{y}|\Theta)$. This is estimated by the maximum likelihood in the MCMC chain. Given a choice of models, the model with the smallest AIC value is considered to be the highest quality model.

4.4.2 BIC

The Bayesian Information Criteria (BIC) [Schwarz1978] is defined as

$$BIC = m_\Theta \ln(d) - 2\ln(L_{max}), \tag{4.12}$$

where d is the number of observations, equal to the length of the array $\boldsymbol{n}$. Given a choice of models, the model with the smallest BIC value is considered to be the highest quality model.

4.4.3 CRPS

The Continuous Ranked Probability Score (CRPS) [Gneiting2007] measures the difference between the CDF of the provided data and that of the forecast/predicted data, i.e., data generated based on the posterior predictive distribution. It is computed by summing up marginal distributions for each day that data is available

$$CRPS = \frac{1}{d}\sum_{j=1}^{d}\int_{-\infty}^{\infty}\left(\mathcal{F}_{pp,j}\left(y_j^{(pp)}|\boldsymbol{y}\right) - \mathcal{H}_{y_j}\left(y_j^{(pp)}\right)\right)^2 dy_j^{(pp)}, \tag{4.13}$$

where $y_j^{(pp)} \equiv y^{(pp)}(t_j)$ is new daily case predictions on day j obtained via the posterior-predictive distribution, $y_j \equiv y(t_j)$ is new daily case data on day j and $\mathcal{F}_{pp,j}$ is the 1-D marginal posterior predictive CDF for day j computed using 1-D marginal posterior predictive distributions

$$\mathcal{F}_{pp,j}(y_j^{(pp)}|\boldsymbol{n}) = \int_{-\infty}^{y_j^{(pp)}} p_{pp,j}\left(y_j^{(pp)'}|\boldsymbol{n}\right) dy_j^{(pp)'} \tag{4.14}$$

where

$$p_{pp,j}\left(y_j^{(pp)}|\boldsymbol{n}\right) = \int p_{pp}(\boldsymbol{y}^{(pp)}|\boldsymbol{y}) d\boldsymbol{y}_{\sim j}^{(pp)} \tag{4.15}$$

is the marginal 1-D posterior predictive density corresponding to day j, based on $p_{pp}\left(\boldsymbol{y}^{(pp)}|\boldsymbol{y}\right)$. Here, $d\boldsymbol{y}_{\sim j}^{(pp)} \equiv dy_1^{(pp)}\cdots dy_{j-1}^{(pp)} dy_{j+1}^{(pp)}\cdots dy_d^{(pp)}$. The CDF of the provided case data $\boldsymbol{y}$ is approximated as a Heaviside function centered at y_j, $\mathcal{H}_{y_j}(y_j^{(pp)}) = 1_{y_j^{(pp)} \geq y_j}$. Like AIC and BIC, the model with the smallest value of CRPS is considered to be of higher quality than other models.

CHAPTER

FIVE

ONE-WAVE MODEL

The following example demonstrates the one-wave model for daily confirmed cases data in the state of New Mexico up to May 13th, 2020, shown in Fig. 5.1.

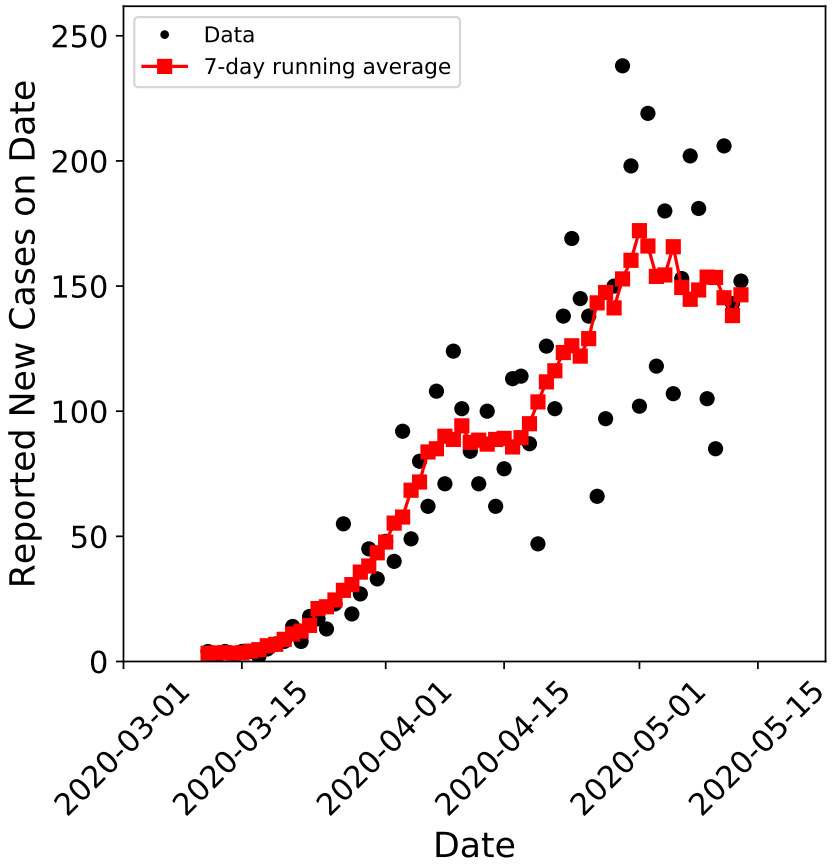

Fig. 5.1: Daily confirmed cases of COVID-19 in New Mexico up to May 13th, 2020. Daily confirmed cases are shown as black symbols, and the corresponding 7-day averaged data shown with red lines and symbols.

5.1 Problem Setup

All the data needed by PRIME is contained in a single json file. The first block of the json file, "regioninfo", should contain information about the data:

```
{
    "regionname":"NM",
    "fchain":"NM_mcmc.h5",
    "day0":"2020-03-01",
    "running_avg_obs":7
}
```

The "regionname" provides the name of the file contain daily confirmed case data. In this case, it is "NM.dat". This file should contain two columns with dates and daily confirmed cases, respectively:

```
2020-03-11 4
2020-03-12 2
2020-03-13 4
2020-03-14 3
2020-03-15 4
2020-03-16 4
2020-03-17 2
2020-03-18 5
2020-03-19 7
2020-03-20 8
2020-03-21 14
2020-03-22 8
2020-03-23 18
2020-03-24 17
```

"fchain" is the name of an HDF5 file containing the MCMC chain along with other useful metadata. The "day0" field specifies the day with the index 0; this is important for setting the prior distribution for t_0. The "running_avg_obs" field sets the number of days to compute a running average over, in this case 7, as shown in Fig. 5.1.

The second block of the json file sets options for the model and MCMC:

```
{
    "model_type": "oneWave",
    "error_model_type": "addMult",
    "logfile":"logmcmcNM.txt",
    "nsteps":1000000,
    "nfinal":10000000,
    "useconv":1,
    "incubation_type":"uncertain",
    "gamma":0.2,
    "splo":[-10,0.0002,0.1, 0.1, 0.0,-20],
    "splhi":[10,0.500,30.0,400.0,10.0,1.0],
    "cini":[0,0.02,6.0,20.2,3.00,0.1],
    "cvini":[0.04,0.001,0.01,0.01,0.01,0.01]
}
```

The first two inputs specify how many waves the model has and the error model. In this example, we use a single infection wave ("oneWave") with the additive and multiplicative error models ("addMult"). The number of steps in the MCMC chain is set by "nsteps" and is 1000000 in this case. The "useconv" option determines if the integrals over probability distributions should be used (1=on). The incubation model is set by "incubation_type", which is set to "uncertain" in this case to model the incubation rate as a random variable instead of a fixed value.

The lists "splo" and "splhi" contain minimum and maximum values, respectively, of model parameters that MCMC

can sample. This overrides the bounds of the prior distributions. The lists "cini" and "cvini" contain initial guesses for the mean and variance of each model parameter.

The prior distributions are specified in the "bayesmod" block:

```
{
    "prior_types":["g","u","u","u","u","u"],
    "prior_info":[[0,1],[0,1],[0,1],[0,1],[0,1],[0,1]]
}
```

The list "prior_types" contains the type of distribution used for each prior. In this case, we are using a Gaussian distribution for the first model parameter, t_0 and uniform distributions for all others. The list "prior_info" contains the mean and standard deviation of each distribution. Note that for uniform distributions, the entry in this list is ignored; the upper and lower bounds are set using entries in "splhi" and "spllo", respectively.

For this case, most of the prior distributions were determined by trial and error, with the exception of the prior for t_0, which was set by observing when the increase in cases started and centering the prior 7-10 days before this to account for incubation time.

Next, the properties of the incubation model are set in the "incopts" section:

```
{
    "incubation_median":5.1,
    "incubation_sigma":0.418,
    "incubation_025":2.2,
    "incubation_975":11.5
}
```

These data are used by PRIME to construct a fixed or uncertain incubation rate model.

A json file containing these section can be used to run the MCMC and output the chain, but other sections are needed for postprocessing. To run PRIME for this case, simply call the "prime_run.py" script followed by the name of the json file.

5.2 New Case Forecast Results

Forecast results can be computed by running the postprocessor script "prime_compute_epi_inf_curves.py" in the same directory as the run. This script requires several additional sections in the json file. Firstly, the "ppopts" section contains the information needed to plot the new case forecast presented in Fig. 5.2.

```
{
     "nstart":100000,
     "nsamples":1000,
     "days_extra":10,
     "runmodel":1,
     "postpred":1,
     "newdata": "NM_future.dat",
     "quantile_newcases":[0.025,0.25,0.5,0.75,0.975],
     "linetype_newcases":["b--","g-","r-","g-","b--"],
     "linewidth_newcases":[3,2,3,2,3],
     "fillbetw_newcases":[[0.25,0.5,"g",0.4],[0.5,0.75,"g",0.4]],
     "xylim_newcases":["2020-03-01","2020-04-15",0,300],
     "xylbl_newcases":["Date",16,"Reported New Cases on Date",16],
     "xyticklbl_newcases":[14,14],
     "newcases":["ko",6],
     "figtype":"pdf",
```

(continues on next page)

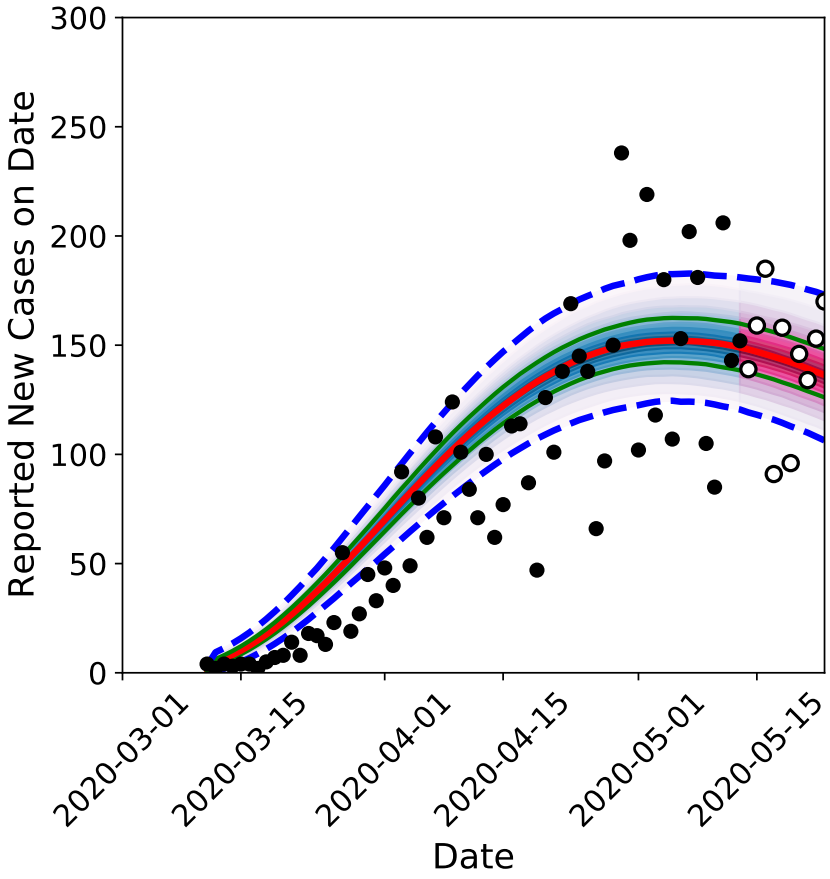

Fig. 5.2: One-wave forecast for New Mexico on May 13th, 2020. Data used to calibrate the epidemiological model are indicated by filled black circles. The shaded color region illustrates either the posterior-predictive distribution with darker colors near the median and lighter colors near the low and high quantile values. The blue colors correspond to the hindcast dates and red colors to forecasts. The inter-quartile range is marked with green lines and the 95% confidence interval with dashed lines. The plot also shows data collected at a later time, with open circles, to check the agreement between the forecast and the observed number of cases after the model has been calibrated.

(continued from previous page)

```
    "fpredout":"NM_epidemic_curve",
    "fout_newcases":"NM_epidemic_curve"
}
```

The portion of the MCMC chain used to generate the plot is set by "nstart" and "nsamples". "nstart" sets the starting index for the portion of the chain used for postprocessing. "nsamples" sets the number of entries in the chain (after index "nstart") to be sampled uniformly for postprocessing.

"days_extra" sets how many days out to compute the forecast, in this case 10 days, or until May 23rd, 2020.

"runmodel" determines whether or not to run the model for each chain sample to compute new cases or to read the new case data from the HDF5 file whose name is specified by the "fpredout" entry in this block. This option should be set to 1 the first time that "prime_compute_epi_inf_curves.py" is run, but can be set to 0 for subsequent runs, for example if one wants to regenerate a plot.

"postpred" is set to 1 to plot the posterior predictive and 0 to plot the push forward PDF.

"newdata" contains the name of an ascii file with future case data. In this case it contains case data from May 14th, 2020 onwards.

The next 8 entires in the "ppopts" block above correspond to plot settings. Many of them concern the quantile curve plots, starting with which quantiles to show ("quantile_newcases"), and the corresponding line color/style ("linetype_newcases"), line width ("linewidth_newcases"), and the color fill between lines ("fillbetw_newcases").

Plot limits, labels, and tick font sizes can be set with "xylim_newcases", "xylbl_newcases", and "xyticklbl_newcases", respectively. Finally, "newcases" contains the a list with the color/symbol followed by the symbol size for the daily new case data used for the forecast.

Finally, "figtype" sets the file format for the forecast plot to written to, "fpredout" contrains the name of an HDF5 file containing the data shown in the forecast plot, and "fout_newcases" is the name of the forecast plot file. The script "prime_compute_epi_inf_curves.py" adds prefixes to indicate which error models are used and if the posterior predictive is plotted. This means that the figure will be written to "NM_newcases_amN_pp.pdf" for our example.

5.3 Infection Rate Prediction Results

To plot the infection rate curve as presented in Fig. 5.3, an "infopts" section is needed in the json file:

```
{
    "inftype":"gamma",
    "ndays":180,
    "runmodel":1,
    "postpred":1,
    "quantile_inf":[0.025,0.25,0.5,0.75,0.975],
    "linetype_inf":["b--","g-","r-","g-","b--"],
    "linewidth_inf":[3,2,3,2,3],
    "fillbetw_inf":[[0.25,0.5,"g",0.4],[0.5,0.75,"g",0.4]],
    "xylim_inf":["2020-03-01","2020-05-01",10,1000],
    "xylbl_inf":["Date",16,"Infection Rate [ppl/day]",16],
    "xyticklbl_inf":[14,14],
    "newcases":["ko",6],
    "figtype":"pdf",
    "finfout":"NM_infection_curve",
    "fout_inf":"NM_infection_curve"
}
```

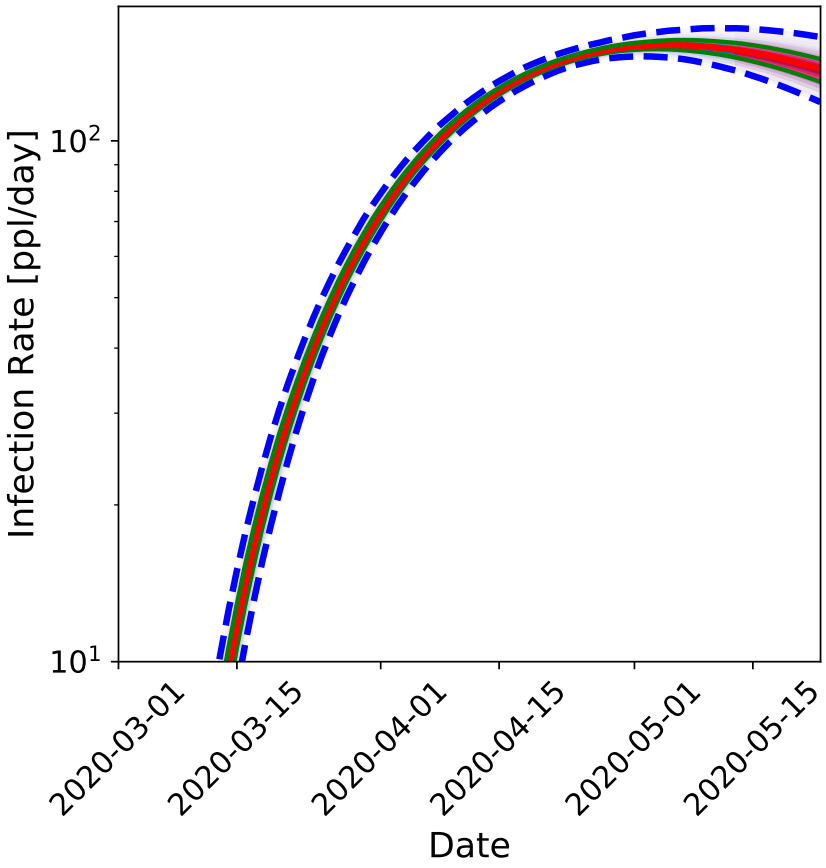

Fig. 5.3: One-wave infection rate curve forecast for New Mexico on May 13th, 2020. The shaded color region illustrates either the posterior-predictive distribution with darker colors near the median and lighter colors near the low and high quantile values. The blue colors correspond to the hindcast dates and red colors to forecasts. The inter-quartile range is marked with green lines and the 95% confidence interval with dashed lines.

Here, "inftype" sets the infection rate curve type, in this case it is a gamma distribution. "runmodel" and "postpred" are the same as in the "ppopts" section. The other entries correspond to the same plot format and file format/name settings in the "ppopts" section.

Finally, new case and infection rate data can be written out in CSV format if a "csvout" section is included in the json file:

```
{
    "nskip":100,
    "finfcurve":"NM_infection_curve",
    "fnewcases":"NM_epidemic_curve",
    "qlist":[0.025,0.25,0.5,0.75,0.975]
}
```

Each row of the CSV files contain data corresponding to each date for which case data is availible along with the dates for which a forecast is availible. In this case, data from early March to May 23rd, 2020 is included. The data on each row includes the date, forecast quantile(s), and individual samples from the MCMC chain. The new case file also contains the reported daily new cases in the last column for all dates in which it is availible. In this case, daily new cases data is included up to May 13th.

"nskip" sets the sampling frequency for the MCMC chain. In this example, every 100th chain sample is included. Recall that the number of chain samples is set to 1000 in the "ppopts" section under "nsamples". This means that the csv files will include 10 samples in this example.

"finfcurve" and "fnewcases" specify the file names with infection rate and new cases data, respectively.

Finally, "qlist" specifies the quantiles for which to output data.

5.4 JSON Input File

The enitre json file is included below for completeness:

```
{
    "regioninfo":{
        "regionname":"NM",
        "fchain":"NM_mcmc.h5",
        "day0":"2020-03-01",
        "running_avg_obs":7
    },
    "mcmcopts":{
        "model_type": "oneWave",
        "error_model_type": "addMult",
        "logfile":"logmcmcNM.txt",
        "nsteps":1000000,
        "nfinal":10000000,
        "useconv":1,
        "incubation_type":"uncertain",
        "gamma":0.2,
        "spllo":[-10,0.0002,0.1, 0.1, 0.0,-20],
        "splhi":[10,0.500,30.0,400.0,10.0,1.0],
        "cini":[0,0.02,6.0,20.2,3.00,0.1],
        "cvini":[0.04,0.001,0.01,0.01,0.01,0.01]
    },
    "bayesmod":{
        "prior_types":["g","u","u","u","u","u"],
        "prior_info":[[0,1],[0,1],[0,1],[0,1],[0,1],[0,1]]
```

(continues on next page)

(continued from previous page)

```
    },
    "incopts":{
        "incubation_median":5.1,
        "incubation_sigma":0.418,
        "incubation_025":2.2,
        "incubation_975":11.5
    },
    "ppopts":{
        "nstart":100000,
        "nsamples":10000,
        "days_extra":10,
        "runmodel":1,
        "postpred":1,
        "newdata": "NM_future.dat",
        "quantile_newcases":[0.025,0.25,0.5,0.75,0.975],
        "linetype_newcases":["b--","g-","r-","g-","b--"],
        "linewidth_newcases":[3,2,3,2,3],
        "fillbetw_newcases":[[0.25,0.5,"g",0.4],[0.5,0.75,"g",0.4]],
        "xylim_newcases":["2020-03-01","2020-04-15",0,300],
        "xylbl_newcases":["Date",16,"Reported New Cases on Date",16],
        "xyticklbl_newcases":[14,14],
        "newcases":["ko",6],
        "figtype":"pdf",
        "fpredout":"NM_epidemic_curve",
        "fout_newcases":"NM_epidemic_curve"
    },
    "infopts":{
        "inftype":"gamma",
        "ndays":180,
        "runmodel":1,
        "postpred":1,
        "quantile_inf":[0.025,0.25,0.5,0.75,0.975],
        "linetype_inf":["b--","g-","r-","g-","b--"],
        "linewidth_inf":[3,2,3,2,3],
        "fillbetw_inf":[[0.25,0.5,"g",0.4],[0.5,0.75,"g",0.4]],
        "xylim_inf":["2020-03-01","2020-05-01",10,1000],
        "xylbl_inf":["Date",16,"Infection Rate [ppl/day]",16],
        "xyticklbl_inf":[14,14],
        "newcases":["ko",6],
        "figtype":"pdf",
        "finfout":"NM_infection_curve",
        "fout_inf":"NM_infection_curve"
    },
    "csvout":{
        "nskip":100,
        "finfcurve":"NM_infection_curve",
        "fnewcases":"NM_epidemic_curve",
        "qlist":[0.025,0.25,0.5,0.75,0.975]
    }
}
```

CHAPTER

SIX

MULTI-WAVE MODEL

The following examples demonstrate the multi-wave capability of PRIME.

6.1 Two Waves

The first example considers daily confirmed case data in the state of New Mexico up to August 26th, 2020, shown in Fig. 6.1.

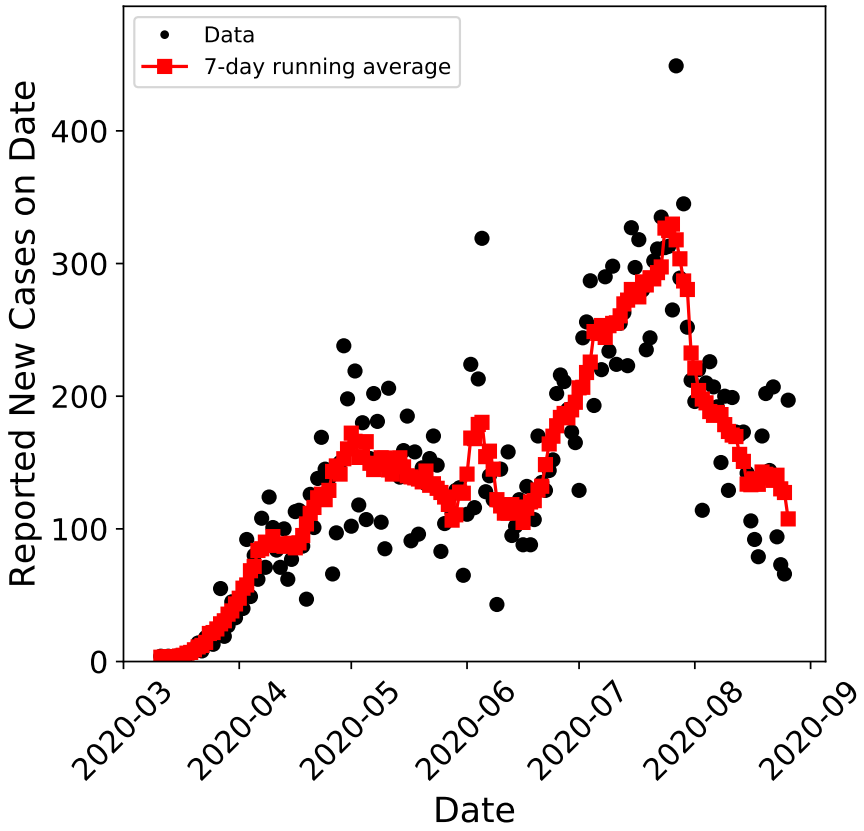

Fig. 6.1: Daily confirmed cases of COVID-19 in New Mexico up to August 26th, 2020. Daily confirmed cases are shown as black symbols, and the corresponding 7-day averaged data shown with red lines and symbols.

We use the two-wave model to fit this since there are two large peaks in new cases in early May and late July 2020.

6.1.1 Problem Setup

While the two wave model can be run independently it is highly recommended to make use of one-wave model results to help compute prior distributions for the first wave in the two-wave model. This is done to i) make use of all available information and ii) limit the parameter range for the early wave parameters to enhance the robustness of the multi-wave model.

The priors for the first wave parameters t_0, N_1, k_1, θ_1 can be then be estimated by the mean and variance of the one-wave MCMC chain. This gives us an informed guess for part of the parameters, limiting the region of parameter space that an MCMC approach needs to explore. A smaller parameter range increases the chance that MCMC is able to find find regions of high likelihood $p\left(\boldsymbol{y}|\Theta\right)$, making the model more robust. The second wave prior distributions should be chosen to include values that approximate the case data well; these distributions may need to be selected by trial and error, although Δt_2 can be estimated by visually inspecting raw daily new case data.

This example uses Gaussian prior distributions for t_0 and Δt_2 and uniform prior distributions for all other model parameters. For Gaussian priors the mean and variance are estimated from the one-wave chain. For uniform priors the minimum and maximum parameter are set equal to $\mu_{chain} \pm 3\sigma_{chain}$, where μ_{chain} and σ_{chain} are the mean and standard deviation computed from the one-wave chain. Note that positive parameters N_1, k_1, θ_1 are restricted to positive values; that is, the lower bound is the maximum of zero and $\mu_{chain} - 3\sigma_{chain}$.

The json file for a two-wave model can be produced from a one-wave model by running the script "prime_compute_prior_from_mcmc.py" in the directory containing the one-wave model json script and HDF5 file with MCMC chain data. The arguments for this script in this case are the name of json file used to run the one-wave model and the model type for the desired json file, in this case "twoWave". For this example, we use the results from example 1 to generate the two-wave json file (included in the "JSON Input Files" section at the end of this example).

It is recommended to experiment with the prior distributions for the 2nd wave parameters Δt_2, N_2, k_2, and θ_2. The time between the first and second wave, Δt_2, can be estimated by the number of days between the first large increase in cases and the second large increase. For this example, we use a wide prior distribution for Δt_2, centered on 60 days with a standard deviation of 15 days.

The two-wave model is also run with the "prime_run.py" script.

6.1.2 New Case Forecast Results

The two-wave forecast results are also computed by running the "prime_compute_epi_inf_curves.py" in the same directory as the run. The postprocessing sections of the json script are the same for multiple waves. Please refer to example 1 for more details on postprocessing.

6.2 Three Waves

The first example considers daily confirmed case data in the state of New Mexico up to November 10th, 2020, shown in Fig. 6.3.

We use the three-wave model to fit this since there are three large peaks in new cases in early May, late July 2020, and cases are rising rapidly in November 2020.

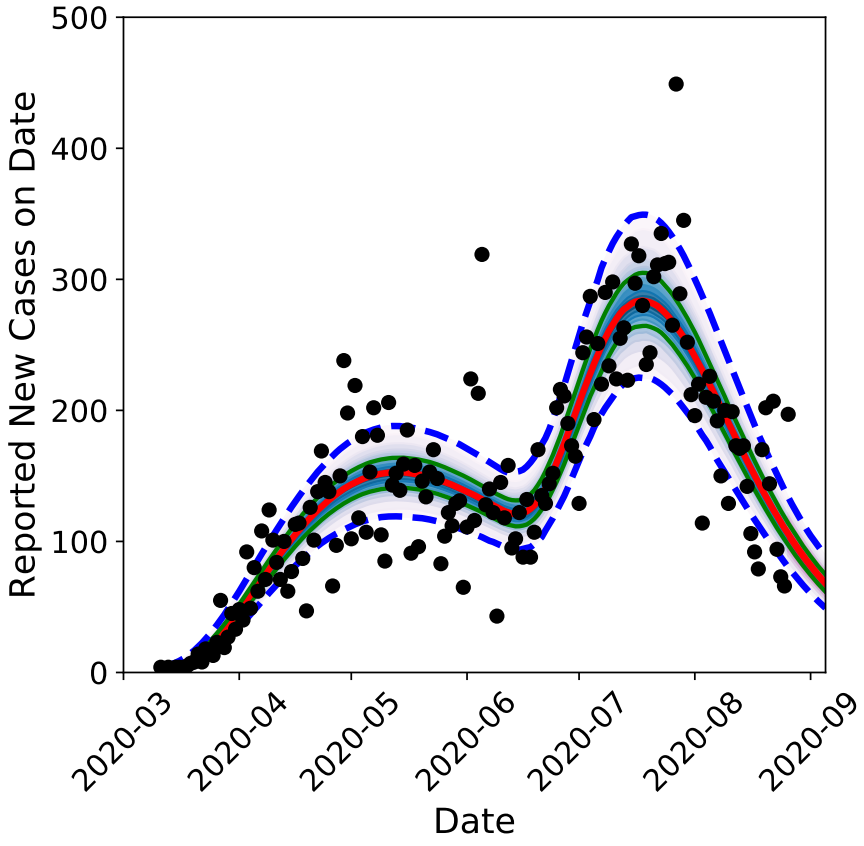

Fig. 6.2: Two-wave forecast for New Mexico on August 26th, 2020. Data used to calibrate the epidemiological model are indicated by filled black circles. The shaded color region illustrates either the posterior-predictive distribution with darker colors near the median and lighter colors near the low and high quantile values. The blue colors correspond to the hindcast dates and red colors to forecasts. The inter-quartile range is marked with green lines and the 95% confidence interval with dashed lines.

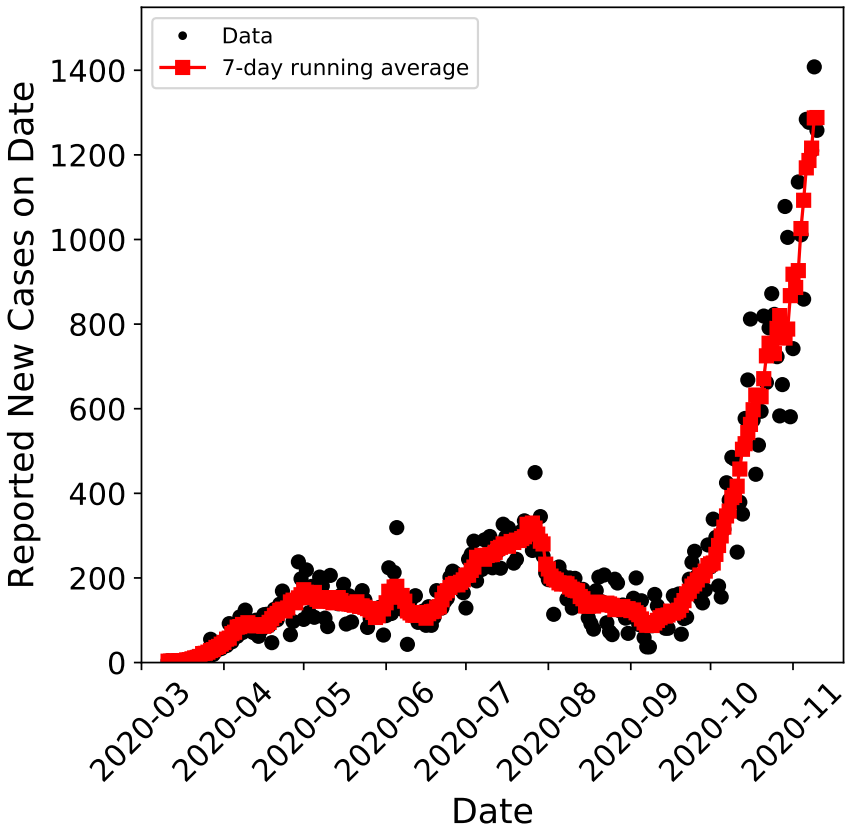

Fig. 6.3: Daily confirmed cases of COVID-19 in New Mexico up to November 10th, 2020. Daily confirmed cases are shown as black symbols, and the corresponding 7-day averaged data shown with red lines and symbols.

6.2.1 Problem Setup

Like the two-wave model, the three-wave model can be run independently but it is highly recommended to make use of two-wave model results to help compute prior distributions for the first and second waves in the three-wave model. The priors for the parameters t_0, N_1, k_1, θ_1, Δt_2, N_2, k_2, and θ_2 are estimated from the two-wave model MCMC chain.

The json file for a three-wave model can be produced from a two-wave model by running the script "prime_compute_prior_from_mcmc.py" in the directory containing the two-wave model json script and HDF5 file with MCMC chain data. The arguments for this script in this case are the name of json file used to run the two-wave model and the model type for the desired json file, in this case "threeWave". For this example, we use the two-wave results from the previous section to generate the three-wave json file (included in the "JSON Input Files" section at the end of this example).

It is recommended to experiment with the prior distributions for the 3rd wave parameters Δt_3, N_3, k_3, and θ_3. The time between the first and third wave, Δt_3, can be estimated by the number of days between the first large increase in cases and the third large increase. For this example, we use a wide prior distribution for Δt_3, centered on 180 days with a standard deviation of 15 days.

The three-wave model is also run with the "prime_run.py" script.

6.2.2 New Case Forecast Results

The three-wave forecast results are also computed by running the "prime_compute_epi_inf_curves.py" in the same directory as the run. The postprocessing sections of the json script are the same for multiple waves. Please refer to example 1 for more details on postprocessing.

6.3 JSON Input Files

6.3.1 Two Wave

```
{
    "regioninfo": {
        "regionname": "NM",
        "fchain": "NM_mcmc.h5",
        "day0": "2020-03-01",
        "running_avg_obs": 7
    },
    "mcmcopts": {
        "model_type": "twoWave",
        "error_model_type": "addMult",
        "logfile": "logmcmcNM_2wave.txt",
        "nsteps": 1000000,
        "nfinal": 10000000,
        "useconv": 1,
        "incubation_type": "uncertain",
        "gamma": 0.7,
        "spllo": [
            -2.877357932637136,
            0.009575447192753876,
            3.4902877959833143,
            12.321758685351922,
            20,
```

(continues on next page)

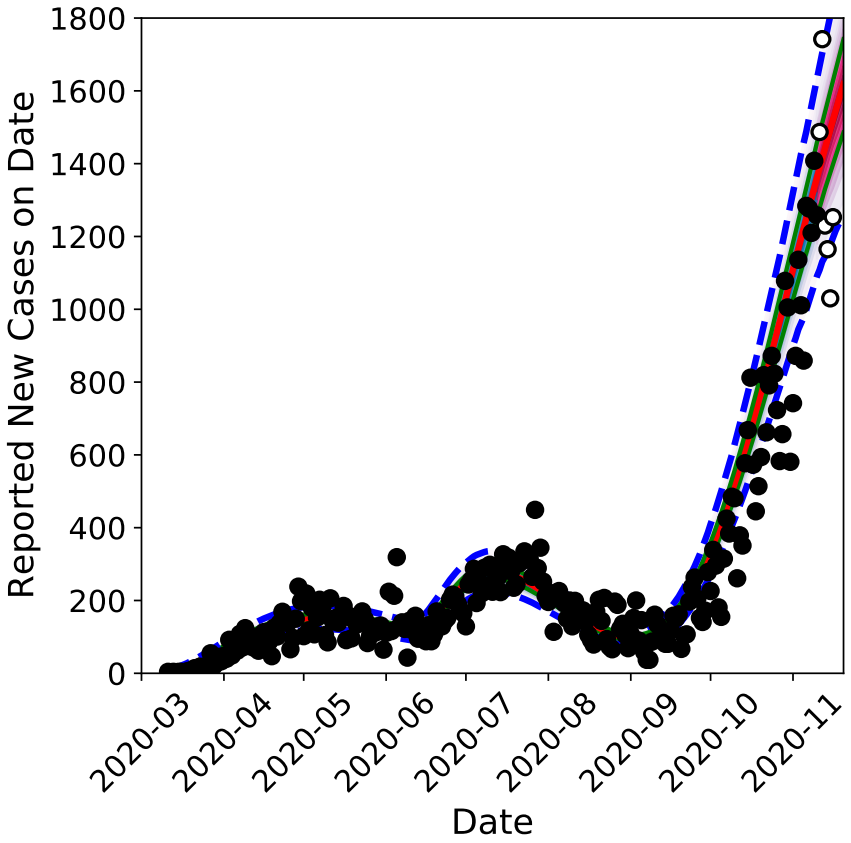

Fig. 6.4: Three-wave forecast for New Mexico on November 10th, 2020. Symbols and lines are the same as in example 1.

(continued from previous page)


```
            0.0002,
            0.1,
            0.1,
            0.0,
            -20
        ],
        "splhi": [
            2.969027281659369,
            0.01839093251179052,
            5.360610614202396,
            26.17367878217896,
            100,
            0.5,
            30.0,
            400.0,
            10.0,
            1.0
        ],
        "cini": [
            0.04583467451111642,
            0.013983189852272197,
            4.425449205092855,
            19.247718733765442,
            60,
            0.02,
            6.0,
            20.2,
            3.0,
            0.1
        ],
        "cvini": [
            0.9494505576095775,
            2.158688372504182e-06,
            0.09716965123197129,
            5.3298802880244684,
            225,
            0.001,
            0.01,
            0.01,
            0.01,
            0.01
        ]
    },
    "bayesmod": {
        "prior_types": ["g","u","u","u","g","u","u","u","u","u"],
        "prior_info": [
            [0.04583467451111642,0.9743975357160841],[0,1],[0,1],[0,1],
            [60,15.0],[0,1],[0,1],[0,1],
            [0,1],[0,1]
        ]
    },
    "incopts": {
        "incubation_median": 5.1,
        "incubation_sigma": 0.418,
        "incubation_025": 2.2,
        "incubation_975": 11.5
    },
```


(continues on next page)

(continued from previous page)

```
    "ppopts": {
        "nstart": 100000,
        "nsamples": 10000,
        "days_extra": 10,
        "runmodel": 1,
        "postpred": 1,
        "quantile_newcases": [0.025,0.25,0.5,0.75,0.975],
        "linetype_newcases": ["b--","g-","r-","g-","b--"],
        "linewidth_newcases": [3,2,3,2,3],
        "fillbetw_newcases": [[0.25,0.5,"g",0.4],[0.5,0.75,"g",0.4]],
        "xylim_newcases": ["2020-03-01","2020-04-15",0,500],
        "xylbl_newcases": ["Date",16,"Reported New Cases on Date",16],
        "xyticklbl_newcases": [14,14],
        "newcases": ["ko",6],
        "figtype": "pdf",
        "fpredout": "NM_epidemic_curve",
        "fout_newcases": "NM_epidemic_curve"
    },
    "infopts": {
        "inftype": "gamma",
        "ndays": 180,
        "runmodel": 1,
        "postpred": 1,
        "quantile_inf": [0.025,0.25,0.5,0.75,0.975],
        "linetype_inf": ["b--","g-","r-","g-","b--"],
        "linewidth_inf": [3,2,3,2,3],
        "fillbetw_inf": [[0.25,0.5,"g",0.4],[0.5,0.75,"g",0.4]],
        "xylim_inf": ["2020-03-01","2020-05-01",10,1000],
        "xylbl_inf": ["Date",16,"Infection Rate [ppl/day]",16],
        "xyticklbl_inf": [14,14],
        "newcases": ["ko",6],
        "figtype": "pdf",
        "finfout": "NM_infection_curve",
        "fout_inf": "NM_infection_curve"
    },
    "csvout": {
        "nskip": 100,
        "finfcurve": "NM_infection_curve",
        "fnewcases": "NM_epidemic_curve",
        "qlist": [0.025,0.25,0.5,0.75,0.975]
    }
}
```

6.3.2 Three Wave

```
{
    "regioninfo": {
        "regionname": "NM",
        "fchain": "NM_mcmc.h5",
        "day0": "2020-03-01",
        "running_avg_obs": 7
    },
    "mcmcopts": {
        "model_type": "threeWave",
        "error_model_type": "addMult",
```

(continues on next page)

(continued from previous page)


```
        "logfile": "logmcmcNM_3wave.txt",
        "method": "am",
        "nsteps": 1000000,
        "nfinal": 10000000,
        "useconv": 1,
        "incubation_type": "uncertain",
        "gamma": 0.7,
        "spllo": [
            -1.5198886917920054,
            0.01343778201509116,
            3.715378521549199,
            17.57101548608055,
            70.81826941872282,
            0.010771637229469904,
            0.0,
            4.976510035388319,
            140,
            0.0002,
            0.1,
            0.1,
            0.0,
            -20
        ],
        "splhi": [
            1.8197861405425901,
            0.016193383448404937,
            4.6929341387263,
            24.574886774303263,
            112.57412043153732,
            0.013487998537736372,
            11.208390968975433,
            15.442199070301406,
            220,
            0.5,
            30.0,
            400.0,
            10.0,
            1.0
        ],
        "cini": [
            0.1499487243752924,
            0.014815582731748048,
            4.204156330137749,
            21.072951130191907,
            91.69619492513007,
            0.012129817883603138,
            5.552822284750233,
            10.209354552844863,
            180,
            0.02,
            6.0,
            20.2,
            3.0,
            0.1
        ],
        "cvini": [
            0.30981744404803085,
```


(continues on next page)

(continued from previous page)

```
            2.1092609053558147e-07,
            0.02654486068540284,
            1.3626170283886236,
            48.43197482789893,
            2.0496163214019776e-07,
            3.5539396824431955,
            3.0425179715416664,
            225,
            0.001,
            0.01,
            0.01,
            0.01,
            0.01
        ]
    },
    "bayesmod": {
        "prior_types": [
            "g","u","u","u",
            "g","u","u","u",
            "g","u","u","u",
            "u","u"
        ],
        "prior_info": [
            [0.1499487243752924,0.5566124720557659],[0,1],[0,1],[0,1],
            [91.69619492513007,6.959308502135749],[0,1],[0,1],[0,1],
            [180,15.0],[0,1],[0,1],[0,1],
            [0,1],[0,1]
        ]
    },
    "incopts": {
        "incubation_median": 5.1,
        "incubation_sigma": 0.418,
        "incubation_025": 2.2,
        "incubation_975": 11.5
    },
    "ppopts": {
        "nstart": 100000,
        "nsamples": 10000,
        "days_extra": 10,
        "runmodel": 1,
        "postpred": 1,
        "quantile_newcases": [0.025,0.25,0.5,0.75,0.975],
        "linetype_newcases": ["b--","g-","r-","g-","b--"],
        "linewidth_newcases": [3,2,3,2,3],
        "fillbetw_newcases": [[0.25,0.5,"g",0.4],[0.5,0.75,"g",0.4]],
        "xylim_newcases": ["2020-03-01","2020-04-15",0,500],
        "xylbl_newcases": ["Date",16,"Reported New Cases on Date",16],
        "xyticklbl_newcases": [14,14],
        "newcases": ["ko",6],
        "figtype": "pdf",
        "fpredout": "NM_epidemic_curve",
        "fout_newcases": "NM_epidemic_curve"
    },
    "infopts": {
        "inftype": "gamma",
        "ndays": 180,
        "runmodel": 1,
```

(continues on next page)

(continued from previous page)

```
        "postpred": 1,
        "quantile_inf": [0.025,0.25,0.5,0.75,0.975],
        "linetype_inf": ["b--","g-","r-","g-","b--"],
        "linewidth_inf": [3,2,3,2,3],
        "fillbetw_inf": [[0.25,0.5,"g",0.4],[0.5,0.75,"g",0.4]],
        "xylim_inf": ["2020-03-01","2020-05-01",10,1000],
        "xylbl_inf": ["Date",16,"Infection Rate [ppl/day]",16],
        "xyticklbl_inf": [14,14],
        "newcases": ["ko",6],
        "figtype": "pdf",
        "finfout": "NM_infection_curve",
        "fout_inf": "NM_infection_curve"
    },
    "csvout": {
        "nskip": 100,
        "finfcurve": "NM_infection_curve",
        "fnewcases": "NM_epidemic_curve",
        "qlist": [0.025,0.25,0.5,0.75,0.975]
    }
}
```

CHAPTER

SEVEN

DEVELOPER REFERENCE GUIDE

7.1 Drivers

source_release.prime_run.**main**(*setupfile*)
Driver script to run MCMC for parameter inference for a multi-wave epidemic model. Currently limited to up to three infection curves.

To run this script:

python <path-to-this-directory>/prime_run.py <name-of-json-input-file>

Parameters

setupfile: string json format input file with information on observations data, filtering options, MCMC options, and postprocessing options. See "setup_template.json" for a detailed example

source_release.prime_plot_data.**main**(*setupfile*)
Plot raw and filtered data for the region specified in the setupfile.

Parameters

setupfile: string json file (.json) including the region name. The "regionname.dat" should exist in the path accessible for this script

source_release.prime_plotKDE.**main**(*filename*)
Plots 1D and 2D marginal kernel density estimates based on MCMC samples

Parameters

filename: string json file (.json) including run setup information and postprocessing information for an MCMC run. It should specify the name of the file containing the MCMC chain

or

pickle file (.pkl) with a dictionary containing the KDE distributions.This file is generated by running this script with a json file (see above)

source_release.prime_compute_info_criteria.**main**(*setupfile*)
This script postprocesses data from PRIME to compute statistical information including: - AIC: Akaike Information Criterion - BIC: Bayesian Information Criterion - CPRS: Continuous Rank Probability Score Results are saved in "info_criteria.txt"

Parameters

setupfile: string json file (.json) including run setup information and postprocessing information for an MCMC run. It should specify the name of the file containing the MCMC chain

source_release.prime_compute_distance_correlation.**main**(*setupfile*)
Computes and saves distance correlations based on samples. The distance correlation matrix is saved in "distanceCorr.txt"

Parameters

setupfile: string json file (.json) including run setup information and postprocessing information for an MCMC run. It should specify the name of the file containing the MCMC chain

7.2 Epidemiological Model

source_release.prime_model.**modelPred**(*state*, *params*, *is_cdf=False*)
Evaluates the PRIME model for a set of model parameters; specific model settings (e.g. date range, other control knobs, etc) are specified via the "params" dictionary

Parameters

state: python list or numpy array model parameters

params: dictionary detailed settings for the epidemiological model

is_cdf: boolean (optional, default False) estimate the epidemiological curve based on the CDF of the incubation model (True) or via the formulation that employs the PDF of the icubation model (False)

Returns

Ncases: numpy array daily counts for people turning symptomatic

source_release.prime_infection.**infection**(*state*, *params*)

Compute infection curve for multi-wave epidemics

- this function is currently used by the post-processing script to push-forward the posterior into a set of infection curves that are consistent with the observed cases

Parameters

state: python list or numpy array model parameters

params: dictionary detailed settings for the epidemiological model

Returns

dates: numpy array list of dates for which the infection rates were computed

infectons: numpy array infection rate values corresponding to the list of dates

source_release.prime_infection.**infection_rate**(*time*, *qshape*, *qscale*, *inftype*)
Infection rate (gamma or log-normal distribution)

Parameters

time: float, list, or numpy array instances in time for the evaluation of the infection_rate model

qshape: float shape parameter

qscale: float scale parameter

inftype: string infection rate type ("gamma" for Gamma distribution, otherwise the Log-normal distribution)

Returns

vals: numpy array infection rates corresponding to the time values provided as input parameters

source_release.prime_incubation.**incubation_fcn**(*time*, *incubation_median*, *incubation_sigma*, *is_cdf=False*)

Computes the incubation rate

Parameters

time: float, list, or numpy array instances in time for the evaluation of the incubation rate model

incubation_median: float median of the incubation rate model

incubation_sigma: float standard deviation of the incubation rate model

is_cdf: boolean (optional, default False) select either the CDF of the incubation rate model (True) or its PDF (False)

Returns

vals: numpy array incubation rates corresponding to the time values provided as input parameters

7.3 Bayesian Inference

source_release.prime_posterior.**logpost**(*state*, *params*)

Compute log-posterior density values; this function assumes the likelihood is a product of independent Gaussian distributions

Parameters

state: python list or numpy array model parameters

params: dictionary detailed settings for the epidemiological model

Returns

llik: float natural logarithm of the likelihood density

lpri: float natural logarithm of the prior density

source_release.prime_posterior.**logpost_negb**(*state*, *params*)

Compute log-posterior density values; this function assumes the likelihood is a product of negative-binomial distributions

Parameters

state: python list or numpy array model parameters

params: dictionary detailed settings for the epidemiological model

Returns

llik: float natural logarithm of the likelihood density

lpri: float natural logarithm of the prior density

source_release.prime_posterior.**logpost_poisson**(*state*, *params*)

Compute log-posterior density values; this function assumes the likelihood is a product of poisson distributions

Parameters

state: python list or numpy array model parameters

params: dictionary detailed settings for the epidemiological model

Returns

llik: float natural logarithm of the likelihood density

lpri: float natural logarithm of the prior density

source_release.prime_mcmc.**ammcmc**(*opts*, *cini*, *likTpr*, *lpinfo*)

Adaptive Metropolis Markov Chain Monte Carlo

Parameters

opts [dictionary of parameters]

- nsteps : no. of mcmc steps
- nburn : no. of mcmc steps for burn-in (proposal fixed to initial covariance)
- nadapt : adapt every nadapt steps after nburn
- nfinal : stop adapting after nfinal steps
- inicov : initial covariance
- coveps : small additive factor to ensure covariance matrix is positive definite (only added to diagonal if covariance matrix is singular without it)
- burnsc : factor to scale up/down proposal if acceptance rate is too high/low
- gamma : factor to multiply proposed jump size with in the chain past the burn-in phase (Reduce this factor to get a higher acceptance rate. Defaults to 1.0)
- spllo : lower bounds for chain samples
- splhi : upper bounds for chain samples
- rnseed : Optional seed for random number generator (needs to be integer >= 0) If not specified, then random number seed is not fixed and every chain will be different.
- tmpchn : Optional; if present, will save chain state every 'ofreq' to ascii file. Filename is randomly generated if tmpchn is set to 'tmpchn', or set to the string passed through this option if not present, chain states are not saved during the MCMC progress

cini [starting mcmc state]

likTpr [log-posterior function; it takes two input parameters as follows]

- first parameter is a 1D array containing the chain state at which the posterior will to be evaluated
- the second parameter contains settings the user can pass to this function; see below info for 'lpinfo'
- this function is expected to return log-Likelihood and log-Prior values (in this order)

lpinfo [info to be passed to the log-posterior function] this object can be of any type (e.g. None, scalar, list, array, dictionary, etc) as long as it is consistent with settings expected inside the 'likTpr' function

Returns

mcmcRes: results dictionary

- 'chain' : chain samples (nsteps x chain dimension)

- 'cmap' : MAP estimate
- 'pmap' : MAP log posterior
- 'accr' : overall acceptance rate
- 'accb' : fraction of samples inside bounds
- 'rejAll' : overall no. of samples rejected
- rejOut' : no. of samples rejected due to being outside bounds
- 'minfo' : meta_info, acceptance probability, log likelihood, log prior
- 'final_cov' : the covariance matrix at the end of the run

7.4 Statistical Utilities

source_release.prime_stats.**computeAICandBIC**(*run_setup*, *verbose=0*)
Compute Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC)

Parameters

run_setup: dictionary with run settings; see the Examples section in the manual

Returns

AIC: float

BIC: float

source_release.prime_stats.**computeCRPS**(*run_setup*)
Compute Continuous Rank Predictive Score (CRPS)

Parameters

run_setup: dictionary with run settings; see the Examples section in the manual

Returns

CRPS: float

source_release.prime_stats.**distcorr**(*spl*)
Compute distance correlation between random vectors

Parameters

spl: numpy array [number of samples x number of variables] first dimension is the number of samples, second dimension is the number of random vectors

Returns

Returns a 2D array of distance correlations between pairs of random vectors; only entries 0<=j<i<no. of random vectors are populated

References: http://en.wikipedia.org/wiki/Distance_correlation

source_release.prime_stats.**getKDE**(*spl*, *nskip=0*, *nthin=1*, *npts=100*, *bwfac=1.0*)
Compute 1D and 2D marginal PDFs via Kernel Density Estimate

Parameters

spl: numpy array MCMC chain [number of samples x number of parameters]

nskip: int number of initial samples to skip when sampling the MCMC chain

nthin: int use every 'nthin' samples

npts: int number of grid points

bwfac: double bandwidth factor

Returns

dict: dictionary with results 'x1D': list of numpy arrays with grids for the 1D PDFs; 'p1D': list of numpy arrays with 1D PDFs; 'x2D': list of numpy arrays of x-axis grids for the 2D PDFs; 'y2D': list of numpy arrays of y-axis grids for the 2D PDFs; 'p2D': list of numpy arrays containing 2D PDFs

7.5 General Utilities

source_release.prime_utils.**compute_error_weight**(*error_info*, *days*)

Compute array with specified weighting for the daily cases data. The weights follow either linear of Gaussian expressions with higher weights for recent data and lower weights for older data

Parameters

error_info: list (error_type,min_wgt,[tau]), error type is either 'linear' or 'gaussian', min_wgt is the minimum weight and tau is the standard deviation of the exponential term if a Gaussian formulation is chosen.

days: int lenght of the weights array

Returns

——-

error_weight: numpy array array of weights

source_release.prime_utils.**prediction_filename**(*run_setup*)

Generate informative name for hdf5 file with prediction data

Parameters

run_setup: dictionary detailed settings for the epidemiological model

Returns

filename: string file name ending with a .h5 extension

source_release.prime_utils.**runningAvg**(*f*, *nDays*)

Apply nDays running average to the input f

Parameters

f: numpy array array (with daily data for this project) to by filtered

nDays: int window width for the running average

Returns

favg: numpy array filtered data

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

[Lauer2020] Lauer S.A., Grantz K.H., Bi Q., Jones F.K., Zheng Q., Meredith H.R., Azman A.S., Reich N.G., Lessler J., The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application, Annals of Internal Medicine (2020),

[Safta2020] Safta C., Ray J., and Sargsyan K., Characterization of partially observed epidemics through Bayesian inference: application to COVID-19, Computational Mechanics (2020),

[JHUCOVID19] COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University

[NYTCOVID19] Coronavirus (Covid-19) Data in the United States

[Akaike1974] Akaike, H., A new look at the statistical model identification, Annals of Statistics (2009)

[Andrieu2009] Andrieu C., Roberts G.O., The pseudo-marginal approach for efficient Monte Carlo computations, Annals of Statistics (2009)

[Gneiting2007] Gneiting T., Raftery A., Strictly Proper Scoring Rules, Prediction, and Estimation (2007)

[Haario2001] Haario H., Saksman E., Tamminen J., An adaptive Metropolis algorithm, Bernoulli (2001)

[Kass1998] Raftery A.E., Lewis S., Markov Chain Monte Carlo in Practice: A Roundtable Discussion, The American Statistician (1998)

[Lloyd2007] Lloyd-Smith J.O., Maximum Likelihood Estimation of the Negative Binomial Dispersion Parameter for Highly Overdispersed Data, with Applications to Infectious Diseases, Public Library of Science (2007)

[Lynch2004] Lynch S.M., Western B., Bayesian posterior predictive checks for complex models, Sociological Methods and Research (2004)

[Raftery1992] Raftery A.E., Lewis S., How Many Iterations in the Gibbs Sampler?, Bayesian Statistics 4 (1992)

[Schwarz1978] Schwarz G., Estimating the Dimension of a Model, Bayesian Statistics 4 (1992)