

Bayesian Modelling Using STAN

Statistical Modelling

- **Traditional methods**

- Small number of well understood methods
- Well-designed laboratory experiments with refutable hypothesis
- Easily interpretable results

- **Real-world analysis**

- Many interacting processes so experimental design is hard
- Very expensive, reuse data as much as possible
- Indirect interpretation of results

- **Statistical Modelling**

- Generative model - mechanistic models of underlying processes (with unknown parameters)
- Observation model - measurement uncertainty and biases (e.g. censoring)
- Very complicated with no standard statistical tests

Stan for Bayesian Modelling

- **What is Stan?**

- package for MCMC sampling
- high-level language for describing Bayesian models

- **Why Stan?**

- very fast development of new models
- computationally and numerically efficient
- integrated with R
- well-documented and well-used with community support

- **State-space model**

- hidden/latent variable models for time-series analysis
- epidemiology model for estimating $R(t)$

https://github.com/BDI-pathogens/stan_epi_tutorial

State-Space Models

- State-space model (a.k.a. filters) are a class of model for analysing time-series data
- The original **Kalman filter** used to model measurement error

$$x_t = y_t + \epsilon_t \quad \text{where} \quad \epsilon_t \sim N(0, \sigma_\epsilon^2) \text{ i.i.d.}$$

$$y_t = y_{t-1} + \mu_t \quad \text{where} \quad \mu_t \sim N(0, \sigma_\mu^2) \text{ i.i.d.}$$

- y_t is the **hidden variable** we are trying to measure
- x_t is the **observed variable** which contains noise
- Aim to decompose the observation into change in the hidden variable and measurement error

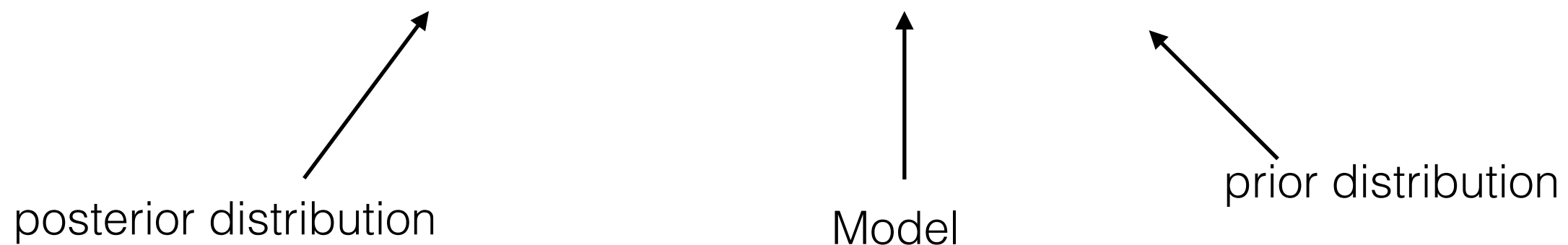
Bayesian Modelling

- Bayes Theorem

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

- Bayesian modelling

$$P(\text{model}|\text{data}) \propto P(\text{data}|\text{model})P(\text{model})$$



- Sample from the posterior distribution using Monte Carlo Markov Chain methods (MCMC)

Epidemiological Model

- Aim - to estimate the reproduction variable with time
- Hidden-variables
 - R_t - the reproduction number

$$R_t = R_{t-1} + \epsilon_t$$
$$\epsilon_t \sim \text{Normal}(0, \sigma_\epsilon)$$

- I_t - the actual number of infections, model with renewal equation

$$I_t = R_t \sum_{\tau} I_{t-\tau} G_{\tau}$$
$$= R_t \sum_{\tau=t_1}^{t_2} \frac{I_{t-\tau}}{t_2 - t_1 + 1}$$

Epidemiological Model

- Observed-variables

- C_t - the observed cases

$$C_t \sim \text{Poisson}(I_t)$$

- Priors

- R_0 - the initial reproduction number

$$R_0 \sim \text{Uniform}(R_{0,\min}, R_{0,\max})$$

- σ_ϵ - the size of daily change in R_t

$$\sigma_\epsilon \sim \text{Uniform}(0, \sigma_{\epsilon,\max})$$

- I_0 - the initial number infections

$$I_0 \sim \text{Uniform}(I_{0,\min}, I_{0,\max})$$

STAN 1.01

- Stan code is split in to blocks which are then interpreted by the Stan compiler to derive a model which can be sampled from. The key ones are:
 - **data** - list all variables which are passed to the model (e.g. t_{\max} , t_1 , $R_{0,\min}$)

```
data {  
  // In the data block all data and constants must be listed with data types  
  // Examples  
  
  // A real variable called r1  
  real r1;  
  
  // A integer variable called n1  
  int<lower=0> n1;  
  
  // An array called my_array of n1 integers  
  array[n1] int my_array;  
}
```


STAN 1.01

- **parameters** - all parameters directly sampled by the model (e.g. R_0 , ε_t)

```
parameters {  
  // All parameters in the model which are sampled directly are listed here  
  // Examples:  
  
  // A real variable called p1 which is between 0 and r1 in value  
  real<lower=0,upper=r1> p1;  
  
  // An array of reals called p_array of n1 values between 0 and r1  
  array[n1] real<lower=0,upper=r1> p_array;  
}
```

STAN 1.01

- **transformed parameters** - quantities derived from the directly sampled parameters (e.g. R_t , I_t)

```
transformed parameters {  
  // Derived variables from the sampled parameters  
  // Examples - a simple random walk  
  
  // An array of reals called walk and length n1  
  array[n1] real walk;  
  
  walk[1] = p1;  
  for( idx in 2:n1 )  
    walk[idx] = walk[idx-1] + p_array[idx];  
}
```

STAN 1.01

- **model** - builds the likelihood of the model given the data (e.g. likelihood of C_t given I_t)

```
model {  
  // Calculate the posterior likelihood of the model given the data  
  
  // p_array is normal distributed with mean 0 and s.d. r1  
  p_array ~ normal( 0, r1 );  
  
  // observed is poisson distributed  
  my_array ~ poisson( walk );  
}
```

Initial Task

- Code the model in Stan and fit to data in R
 - download project from GitHub: `BDI-pathogens/stan_epi_tutorial`
 - install R packages: - `hds_dtc/install.R`
 - template Stan script: - `hds_dtc/simple_model.stan`
 - template R: - `hds_dtc/simple_model.R`
- Pointers
 - put ranges on all parameters (i.e. the priors)
 - need to have infections for t_2 days prior to the first observed point, set all to be I_0

Investigations

- Explore fitting the model to investigate
 - try the different simulated data files, what is the difference, do you need to change priors?
 - `hds_dtc/data/task1a.Rdata`
 - `hds_dtc/data/task1b.Rdata`
 - `hds_dtc/data/task1c.Rdata`
 - effect of the generation time on estimates of $R(t)$
 - censored data - how does the estimate of $R(t)$ change depend upon the length of the data window

Working code is `hds_dtc/task1.R`, please try first, or just use it for hints