Intro to rstan

Ian Laga

8/3/2019

Example 1: Truncated Exponential

For this example, we know the data come from an exponential distribution with rate λ, but we don't observe observations below L or above U (for this example, we know these truncation points). So, the distribution is specified by

$$\lambda \sim \pi_0(\theta)$$

$$Y|\lambda \sim Exp(\lambda)I(L < \lambda < U)$$

For now, we will consider a flat prior on λ. The model is specified in the stan file by

```
// The input data is a vector 'y' of length 'N'.
data {
  int<lower=0> N;
  real L;
  real U;
  real<lower=L,upper=U> y[N];
}

parameters {
  real<lower=0> lambda;
}

// The model to be estimated. We model the output
// 'y' to be exponentially distributed with rate 'lambda'
model {
  // If you wanted a prior, it would go here
  for (n in 1:N)
   y[n] ~ exponential(lambda) T[L,U]; # The truncation occurs between L and U
}
```

Example 2: Censored Exponential

This is similar to the truncated exponential, but now we know how many observations were censored. To incorporate these censored observations, we sample them as latent variables.

```
data {
  int<lower=0> N_obs;
  int<lower=0> N_cens;
  real y_obs[N_obs]; // vector[N]<lower = L, upper = U> foo
  real<lower=max(y_obs)> U;
}

parameters {
  real<lower=U> y_cens[N_cens];
  real<lower=0> lambda;
}

model {
  y_cens ~ exponential(lambda);
  y_obs ~ exponential(lambda);
}
```

Notice that we now have two likelihoods in the model part of the code and two parameters. Everything in the parameter section will be sampled from the posterior distribution.

Example 3: Censored Data with Covariates

We are extending the above example, except now,

$$\lambda = \beta X,$$

where X is known. We have to specify a few more data components now.

```
data {
  int<lower=0> N_obs;
  int<lower=0> N_cens;
  int<lower=0> K;
  matrix[N_obs, K] x_obs;
  matrix[N_cens, K] x_cens;
  real y_obs[N_obs];
  real<upper=min(y_obs)> U;
}

parameters {
  vector<lower = 0>[K] beta;
  real<lower = 0,upper=U> y_cens[N_cens];
}

model {
  y_cens ~ exponential(x_cens * beta);
  y_obs ~ exponential(x_obs * beta);
}
generated quantities {
  vector[K] dif = beta - beta[1];
}
```

Example 4: Linear Regression

This is just very similar to example 3, so I won't discuss it here.

Example 5: Historical Data

This will be the most complicated model we consider, by far. The model for including historical control data that I am considering is from Brard et al. (2019). In words, the prior distribution for the control arm is a power prior, which raising the likelihood of the historical data to power α_0, which controls the effect of the historical data. $\alpha_0 = 0$ means we ignore the historical data, while $\alpha_0 = 1$ means we weight the historical data evenly with the likelihood of the new control trial data. The model for the control arm is

$$\pi(\theta|D_C^H, \alpha_0) \propto L(\theta|D_C^H)^{\alpha_0}\pi_0(\theta)$$

For the treatment effect, we are also considering information from historical aggregate data. The treatment effect is specified by β, and is a weighted sum of the existing information (D_{TE}^H), and another prior.

$$\pi(\beta|D_{TE}^H, \omega) = \omega \times \pi_H(\beta|D_{TE}^H) + (1-\omega) \times \pi_0(\beta)$$

So, assuming independence between the control arm parameters θ and the treatment effect β, the joint prior is just the product of the above two priors, i.e.

$$\pi(\theta, \beta|D_C^H, D_{TE}^H, \alpha_0, \omega) \propto L(\theta|D_C^H)^{\alpha_0}\pi_0(\theta) \times [\omega \times \pi_H(\beta|D_{TE}^H) + (1-\omega) \times \pi_0(\beta)]$$

For this example, the likelihood function for the data is the Weibull distribution with parameters shape α and scale $= \gamma$. The treatment effect is on γ, so the distribution of the treatment arm follows a Weibull distribution with α and $\gamma + \beta$. (Note, depending on the parameterization, the names are changed and inverses can be taken)

I follow the same choices for $\pi_0(\beta)$, etc, that the paper makes. So,

$$\pi_H(\beta|D_{TE}^H) \sim N(\mu_H, \sigma_H)$$

$$\pi_0(\beta) \sim N(0, \sqrt{10})$$

$$\pi_0(\theta) \sim N(0, 10)$$

```
data {
  int<lower=0> N_c;
  int<lower=0> N_t;
  int<lower=0> N_h;
  real<lower=0, upper=1> alpha_0;
  real<lower=0,upper=1> w;
  real mu_h;
  real<lower=0> sigma_h;
  vector[N_h] y_hist;
  vector[N_t] y_treat;
  vector[N_c] y_cont;
}


parameters {
  real<lower=0> beta_0;
  real<lower=0> gamma;
  real beta;
```

```
}


model {
  target += alpha_0 * weibull_lpdf(y_hist | beta_0, gamma) +
            normal_lpdf(beta_0 | 0, 100) + inv_gamma_lpdf(gamma | 0.0001, 0.0001) +
            log(w * exp(normal_lpdf(beta | mu_h, sigma_h)) +
                (1 - w) * exp(normal_lpdf(beta | 0, sqrt(10)))) +
                // Everything up to here is the joint prior
            weibull_lpdf(y_treat | beta_0, gamma + beta) + weibull_lpdf(y_cont | beta_0, gamma);
}
```

Example 6: Custom Function

Here, we are going to consider a regular exponential distribution, but we are going to parameterize the log-likelihood ourselves.

```
functions {
  real exp_mean_lpdf(vector x, real lambda) {
    vector[num_elements(x)] lprob;
    lprob = log(1/lambda) - x/lambda;
    return sum(lprob);
  }
}
data {
  int<lower=0> N;
  vector[N] y;
}

parameters {
  real<lower=0> lambda;
}

model {
  // target += log(1/lambda) - y / lambda;  // You can do it without the function
  // y ~ exp_mean(lambda); // Or you can use the function
  target += exp_mean_lpdf(y | lambda); // This is equivalent to the line above

  // notice that it doesn't have _lpdf when using ~ notation
}
```