

Fast prototyping of Bayesian models using Stan

Claas Heuer

September, 2018

STgenetics

Outline

- Concept
- Examples
- Potential Applications

Concept

Concept

- A modelling language allows definition of (non) linear hierarchical models using descriptive API
- Examples include *WinBUGS*, *Jags* and *Stan*
- Main usecases: Education, Prototyping, very flexible Bayesian inference
- Modelling languages **empower** researchers to run the models they want instead of those implemented in statistical software

- At its core, Stan models define log density functions of the (constrained) parameters
- This includes priors and hyperpriors
- Full Bayesian inference through *Euclidean Hamiltonian Monte Carlo* (HMC / NUTS)
- Supposedly relatively low autocorrelation and fast convergence to equilibrium distribution
- Alternatively: Penalized Maximum Likelihood point estimates of parameters through optimization

Concept

A stan program is parameterized into different segments:

```
functions {  
    // ... function declarations and definitions ...  
}  
data {  
    // ... declarations ...  
}  
transformed data {  
    // ... declarations ... statements ...  
}  
parameters {  
    // ... declarations ...  
}  
transformed parameters {  
    // ... declarations ... statements ...  
}  
model {  
    // ... declarations ... statements ...  
}  
generated quantities {  
    // ... declarations ... statements ...  
}
```

Data

```
data {  
  int<lower=0> Px;  
  int<lower=0> N;  
  matrix[N,Px] X;  
  real y[N];  
}
```

Parameters

```
parameters {  
  vector[Px] beta; // flat prior from  $-\text{Inf}$  to  $\text{Inf}$   
  real<lower=0> sigma; // truncate support  
  real<lower=0> tau; // priors come later  
}
```


Model

```
model {  
  y ~ normal(X * beta , sigma); // the likelihood  
  sigma ~ cauchy(0,5); // cauchy prior on sigma  
}
```

Examples

Mixture of K normals

$$p(\mathbf{x}) = \sum_{i=1}^K \lambda_i f_i(\mathbf{x}), \quad \text{with } \sum \lambda = 1. \quad (1)$$

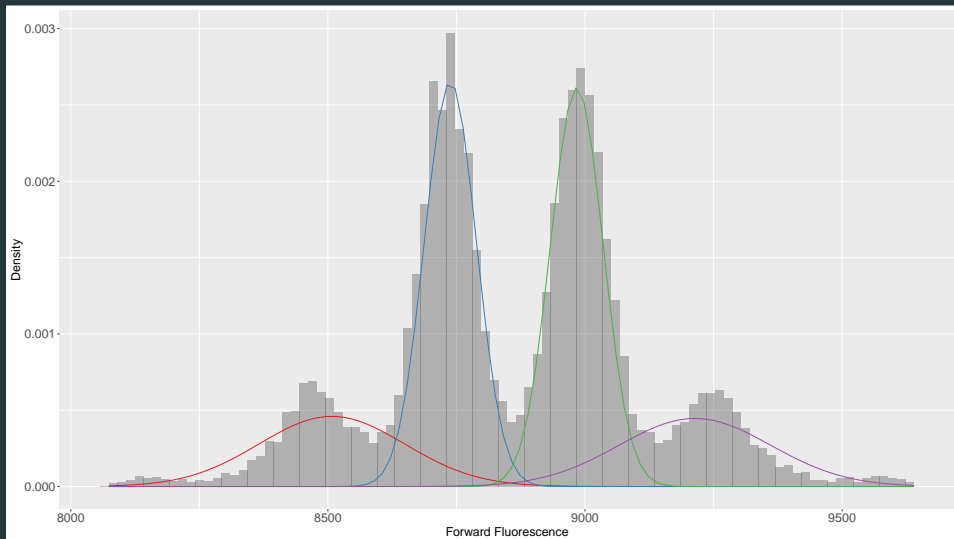
Univariate normal distribution: $f(x) \sim N(\mu, \sigma^2)$

- μ is a single location parameter - the mean
- σ^2 is a common variance component

Model

```
data {  
  int<lower=1> N;  
  int<lower=1> k;  
  real y[N];  
}  
  
parameters {  
  simplex[k] theta;  
  real mu[k];  
  real<lower=0> tau[k];  
}  
  
model {  
  real ps[k];  
  for (i in 1:k){  
    mu[i] ~ normal(0, 1.0e+2);  
  }  
  for(i in 1:N){  
    for(j in 1:k){  
      ps[j] <- log(theta[j]) + normal_log(y[i], mu[j], tau[j]);  
    }  
    increment_log_prob(log_sum_exp(ps));  
  }  
  tau ~ cauchy(0,5);  
}
```

Mixture of 4 normals



Multivariate Ridge Regression

$$\mathbf{Y} = \mathbf{XB} + \mathbf{ZU} + \mathbf{E},$$

$$\mathbf{Y} \sim \text{MVN}(\mathbf{XB} + \mathbf{ZU}, \boldsymbol{\Sigma}_E)$$

$$\mathbf{B} \sim \mathbf{U}(-\text{Inf}, \text{Inf})$$

$$\mathbf{U} \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_U)$$

$$\boldsymbol{\Sigma}_U = \text{diag}(\mathbf{S})_U \boldsymbol{\Omega}_U \text{diag}(\mathbf{S})_U$$

$$\mathbf{S}_U \sim \text{Cauchy}(\mathbf{0}, 5)$$

$$\boldsymbol{\Omega}_U \sim \text{LKJ}(1)$$

$\boldsymbol{\Sigma}_E$ analog to $\boldsymbol{\Sigma}_U$

Model

```
model {  
  // MVN likelihood  
  for(i in 1:N) Y[i,] ~ multi_normal_cholesky(X[i,] * B + Z[i,] * U,  
    diag_pre_multiply(sigmaE, OmegaCholE));  
  
  // sample random effects from mvn  
  for(i in 1:PzU) U[i,] ~ multi_normal_cholesky(musU, diag_pre_multiply(sigmaA, OmegaCholU));  
  
  // cauchy prior on the standard deviations  
  sigmaE ~ cauchy(0, 5);  
  sigmaU ~ cauchy(0, 5);  
  
  // LKJ prior on the correlation matrices (1 is uniform in the interval -1 to 1)  
  OmegaCholE ~ lkj_corr_cholesky(1);  
  OmegaCholU ~ lkj_corr_cholesky(1);  
}
```

MCMC diagnostics



Figure 2: MCMC diagnostics and posterior estimation with shinystan

BayesC π threshold model - Jags

$$\mathbf{y} \sim \mathbf{B}(1, \mathbf{p})$$

$$\mathbf{p} = \text{probit}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u})$$

$$\boldsymbol{\beta} \sim \mathbf{U}(-\text{Inf}, \text{Inf})$$

$$\mathbf{u} \sim \pi N(0, \sigma_u^2) + (1 - \pi)\delta_0$$

$$\pi \sim U(0, 1)$$

$$\sigma_u^2 \sim \text{Gamma}(1, 0.01)$$

BayesC π threshold model - Jags

```
model {  
  for (i in 1:n) {  
    y[i] ~ dbern(prob[i])  
    probit(prob[i]) <- alpha + inprod(X[i,], beta)  
  }  
  for (j in 1:p) {  
    ind[j] ~ dbern(pind)  
    betaT[j] ~ dnorm(0, tau)  
    beta[j] <- ind[j] * betaT[j]  
  }  
  tau ~ dgamma(1, 0.01)  
  alpha ~ dnorm(0, 0.0001)  
  pind ~ dunif(0, 1)  
}
```

BayesC π threshold model - Jags

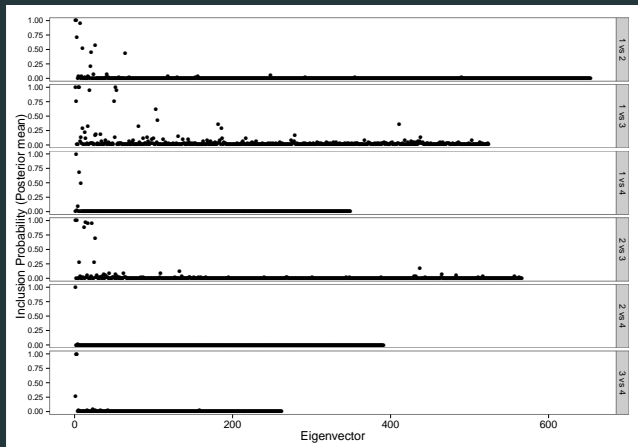


Figure 3: Inclusion probabilities of eigenvectors from a BayesC π threshold model (Heuer et al., 2016)

Potential Applications

Bayesian Alphabet

Table 1. Summary of some effect size distributions that have been proposed for polygenic modeling.

Effect Size Distribution		Keywords	Selected References
Name	Formula		
t	$\beta_i \sim t(0, \nu, \sigma_a^2)$	BayesA	[27,32,33,40]
point-t	$\beta_i \sim \pi t(0, \nu, \sigma_a^2) + (1 - \pi)\delta_0$	BayesB, BayesD, BayesD π	[27,32–34,40]
t mixture	$\beta_i \sim \pi t(0, \nu, \sigma_a^2) + (1 - \pi)t(0, \nu, 0.01\sigma_a^2)$	BayesC	[32,33]
point-normal	$\beta_i \sim \pi N(0, \sigma_a^2) + (1 - \pi)\delta_0$	BayesC, BayesC π , BVSR	[18,19,34]
double exponential	$\beta_i \sim \text{DE}(0, \theta)$	Bayesian Lasso	[28,39,68]
point-normal mixture	$\beta_i \sim \pi_1 N(0, \sigma_a^2) + \pi_2 N(0, 0.1\sigma_a^2) + \pi_3 N(0, 0.01\sigma_a^2) + (1 - \pi_1 - \pi_2 - \pi_3)\delta_0$	BayesR	[35]
normal	$\beta_i \sim N(0, \sigma_a^2)$	LMM, BLUP, Ridge Regression	[22,26,28,48]
normal-exponential-gamma	$\beta_i \sim \text{NEG}(0, \kappa, \theta)$	NEG	[16]
normal mixture	$\beta_i \sim \pi N(0, \sigma_a^2 + \sigma_b^2) + (1 - \pi)N(0, \sigma_b^2)$	BSLMM	Present Work

The reference list contains only a selection of relevant publications. Abbreviations: DE denotes double exponential distribution, NEG denotes normal exponential gamma distribution, and other abbreviations can be found in the main text. In the scaled t-distribution, ν and σ_a^2 are the degree of freedom parameter and scale parameter, respectively. In the DE distribution, θ is the scale parameter. In the NEG distribution, κ and θ are the shape and scale parameters, respectively. Notes: 1. Some applications of these methods combine a particular effect size distribution with a random effects term, with covariance matrix K , to capture sample structure or relatedness. If $K \propto XX^T$ then this is equivalent to adding a normal distribution to the effect size distribution. The listed effect size distributions in this table do not include this additional normal component. 2. BayesC has been used to refer to models with different effect size distributions in different papers. 3. In some papers, keywords listed here have been used to refer to fitting techniques rather than effect size distributions.

doi:10.1371/journal.pgen.1003264.t001

Figure 4: Priors on marker effects from the Bayesian Alphabet (Zhou et al., 2013)

Bayesian Alphabet

The screenshot shows the GitHub interface for the repository `cheuerde / BayesianAlphabet`. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name is displayed with buttons for Unwatch (1), Star (1), and Fork (0). A secondary navigation bar includes links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The main description states: "Implementation of Bayesian Regression models from the 'Bayesian Alphabet' family in Stan and Jags." Below the description, statistics show 9 commits, 1 branch, 0 releases, and 1 contributor. Action buttons include "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download". A table of commit history is shown below, listing files like `BayesCJags.r`, `BivariateMarkerRegressionStan.r`, `JagsBayesR.r`, `LKJCholeskyPriorMultivariateMarkerRe...`, `README.md`, and `StanBayesianRegression.r` along with their commit messages and dates.

cheuerde / **BayesianAlphabet** Unwatch 1 ★ Star 1 🍴 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Implementation of Bayesian Regression models from the "Bayesian Alphabet" family in Stan and Jags. Edit

Add topics

9 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

cheuerde	Create rr_stan.r	Latest commit f569ddb on Aug 12
BayesCJags.r	initial commit	2 years ago
BivariateMarkerRegressionStan.r	Create BivariateMarkerRegressionStan.r	a month ago
JagsBayesR.r	initial commit	2 years ago
LKJCholeskyPriorMultivariateMarkerRe...	Create LKJCholeskyPriorMultivariateMarkerRegressionStan.r	a month ago
README.md	Update README.md	2 years ago
StanBayesianRegression.r	Update StanBayesianRegression.r	2 years ago

<https://github.com/cheuerde/BayesianAlphabet>

Potential Applications

- Fast prototyping of Bayesian models
- Replication of research results
- Experimenting with novel models, e.g. skewed likelihoods, multivariate extensions, alternative priors, etc
- Education and labs
- Packages available for R, julia, Python, etc
- R: *brms* most powerful (non) linear model package to date
- Stan is under active development with large user base
- StanCon in Helsinki just concluded

Questions?
