Some Examples

May 30, 2017

Example 1

- Suppose we want to generate *n* random points in the circle of radius 1 centered at the origin, conditional on the event that no two points are within a distance *d* of each other,where $\beta = P(\text{no two points are within d of each other})$ is assumed to be a small positive number.
- We generate the points with Gibbs sampling.

Example 1

Algorithm:

- Generate n points in the circle randomly.
- Generate a random number U, and let $I = int(nU) + 1$,
- Generate a random point in the circle.
- If this point is not within d of any other $n - 1$ points excluding x_I, then replace x_I by this point; otherwise, generate a new point and repeat the process.

Example 2

- Let $X_i, i = 1, 2, 3, 4, 5$, be independent exponential random variables, with X_i having mean i, and suppose we are interested in using simulation to estimate

$$\beta = P\{\Pi_{i=1}^{5} X_i > 120 | \sum_{i=1}^{5} X_i = 15\}$$

Example 2

- Suppose X and Y are independent exponentials with respective rates λ and μ, where $\mu < \lambda$. The conditional distribution of X given $X + Y = a$ is:

$$\begin{aligned} f_{X|X+Y}(x|a) &= C_1 f_{X,Y}(x, a - x), \\ &= C_2 e^{-\lambda x} e^{-\mu(a-x)}, \\ &= C_3 e^{-(\lambda-\mu)x}, 0 < x < a \end{aligned}$$

Example 2

Algorithm:

- Start with $(x_1, x_2, x_3, x_4, x_5)$ with $x_1 + x_2 + x_3 + x_4 + x_5 = 15$.
- Randomly choose two elements from $\{1, 2, 3, 4, 5\}$, say $I = 2$, $J = 5$.
- Generate the random variables x_2, x_5 with mean 2 and 5, given their sum $x_2 + x_5 = 15 - x_1 - x_3 - x_4$. This is generated by generating the value of an exponential with rate $1/2 - 1/5 = 3/10$, and set x_2 equal to that value and reset x_5 to make $x_1 + x_2 + x_3 + x_4 + x_5 = 15$.
- Repeat the process and the proportion of $\Pi_{i=1}^{5} X_i > 120$ is the estimate of β.

Example 3: Linkage problem

- 197 animals are distributed into 4 categories $Y = (Y_1, Y_2, Y_3, Y_4)$ according to the genetic Linkage model: $p_1 = (2 + \phi)/4, p_2 = (1 - \phi)/4, p_3 = (1 - \phi)/4, p_4 = \phi/4$
- $(Y_1, Y_2, Y_3, Y_4) \sim \mathit{multinomial}(n = 197, p_1, p_2, p_3, p_4)$
- Under the prior:$\phi \sim \mathit{beta}(a, b)$, find the posterior mode.

EM method

- Multinomial distribution:

$$P(X_1 = n_1, \cdots, X_k = n_k) = \frac{n!}{n_1!n_2!\cdots n_k!} p_1^{n_1} \cdots p_k^{n_k}, \quad \sum_{i=1}^{k} n_i = n$$

- The observed data vector of frequencies:

$$y = (y_1, y_2, y_3, y_4)^T.$$

- Suppose the data are generated from a multinomial distribution with probabilities:

$$\frac{1}{2} + \frac{1}{4}\phi, \frac{1}{4}(1-\phi), \frac{1}{4}(1-\phi), \frac{1}{4}\phi$$

- Use MLE to estimate ϕ.

- The probability function is:

$$L(\phi; y) = \frac{n!}{y_1!y_2!y_3!y_4!}(\frac{1}{2} + \frac{1}{4}\phi)^{y_1}(\frac{1}{4}(1-\phi))^{y_2}(\frac{1}{4}(1-\phi))^{y_3}(\frac{1}{4}\phi)^{y_4}$$

- The log likelihood function apart from an additive term not involving ϕ is:

$$\log L(\phi) = y_1 \log(2+\phi) + (y_2 + y_3)\log(1-\phi) + y_4 \log\phi$$

- $$\frac{\partial \log L(\phi)}{\partial\phi} = \frac{y_1}{2+\phi} - \frac{y_2+y_3}{1-\phi} + \frac{y_4}{\phi}$$

$$I(\phi; y) = -\partial^2 \log L(\phi)/\partial\phi^2 = \frac{y_1}{(2+\phi)^2} + \frac{y_2+y_3}{(1-\phi)^2} + \frac{y_4}{\phi^2}$$

- Suppose $y_1 = y_{11} + y_{12}$, where y_{11} and y_{12} have probabilities $\frac{1}{2}$ and $\frac{1}{4}\phi$.
- Suppose y_{11}, y_{12} are unobservable, we only observe their sum y_1. Then the observed vector of frequencies y is viewed as being incomplete and the complete-data vector is taken to be

  $$x = (y_{11}, y_{12}, y_2, y_3, y_4)^T.$$

  They are assumed to arise from a multinomial distribution with probabilities

  $$\frac{1}{2}, \frac{1}{4}\phi, \frac{1}{4}(1-\phi), \frac{1}{4}(1-\phi), \frac{1}{4}\phi$$

  The log likelihood for the complete-data is:

  $$\log L_c(\phi) = (y_{12} + y_4)\log\phi + (y_2 + y_3)\log(1-\phi)$$

- Use MLE, we will get $\phi = \frac{y_{12}+y_4}{y_{12}+y_2+y_3+y_4}$.
- Since the frequency y_{12} is unobservable, we are unable to estimate ϕ.
- We can use an iterative method to estimate ϕ.
- For unobserved data, we fill in by averaging the complete-data log likelihood over its conditional distribution given the observed data y.

- Given a specified value of ϕ^0, the conditional expectation of $\log L_c(\phi)$ can be written as:

$$Q(\phi;\phi^0) = E_{\phi^0}\{\log L_c(\phi)|y\}$$

- As $\log L_c(\phi)$ is a linear function of y_{11}, y_{12}, we can replace y_{12} by its current conditional expectation given the observed data y.
- The random variable Y_{11} corresponding to y_{11} has a binomial distribution with sample size y_1 and probability parameter $\frac{1}{2}/(\frac{1}{2}+\frac{1}{2}\phi^0)$.

$$E_{\phi^0}(Y_{11}|y_1) = y_{11}^0 = \frac{1}{2}y_1/(\frac{1}{2}+\frac{1}{4}\phi^0)$$

$$y_{12}^0 = y_1 - y_{11}^0 = \frac{1}{4}y_1\phi^0/(\frac{1}{2}+\frac{1}{4}\phi^0)$$

M-step: Maximization Q, we get

$$\phi^1 = \frac{y_{12}^0 + y_4}{y_{12}^0 + y_2 + y_3 + y_4}$$

Iteration steps:

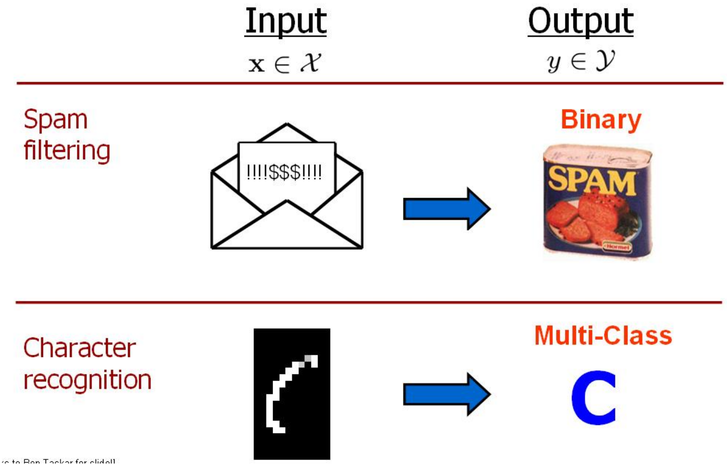
$$\phi^{k+1} = (y_{12}^k + y_4)/(n - y_{11}^k)$$

where

$$y_{11}^k = \frac{1}{2} y_1 / (\frac{1}{2} + \frac{1}{4}\phi^k)$$

$$y_{12}^k = y_1 - y_{11}^k$$

Example:Logistic regression



Input

$\mathbf{x} \in \mathcal{X}$

Output

$y \in \mathcal{Y}$

Spam filtering

!!!!$$$!!!!

Binary

Character recognition

Multi-Class

C

ks to Ben Taskar for slide!]

Logistic Regression

Preserve linear classification boundaries.

- By the Bayes rule:

$$\hat{G}(x) = arg \max_k P(G = k|X = x).$$

- Decision boundary between class k and l is determined by the equation:

$$Pr(G = k|X = x) = Pr(G = l|X = x)$$

.

- Divide both sides by $Pr(G = l|X = x)$ and take log. The above equation is equivalent to

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = 0$$

- Since we enforce linear boundary, we can assume

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = a_0^{(k,l)} + \sum_{j=1}^{p} a_j^{(k,l)} x_j$$

- For logistic regression, there are restrictive relations between $a(k, l)$ for different pairs of (k, l).

Assumptions

$$\log \frac{Pr(G = 1|X = x)}{Pr(G = K|X = x)} = \beta_{10} + \beta_1^T x$$

$$\log \frac{Pr(G = 2|X = x)}{Pr(G = K|X = x)} = \beta_{20} + \beta_2^T x$$

$$\log \frac{Pr(G = K-1|X = x)}{Pr(G = K|X = x)} = \beta_{(K-1)0} + \beta_{K-1}^T x$$

- For any pair (k, l):

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = \beta_{k0} - \beta_{l0} + (\beta_k - \beta_l)^T x.$$

- Number of parameters: $(K - 1)(p + 1)$.
- Denote the entire parameter set by

$$\theta = \{\beta_{10}, \beta_1, \beta_{20}, \beta_2, \cdots, \beta_{(K-1)0}, \beta_{K-1\cdot}\}$$

- The log ratio of posterior probabilities are called log-odds or logit transformations.

- Under the assumptions, the posterior probabilities are given by:

$$Pr(G = k|X = x) = \frac{exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{l=1}^{K-1} exp(\beta_{l0} + \beta_l^T x)}$$

  for $k = 1, ..., K - 1$.

$$Pr(G = K|X = x) = \frac{1}{1 + \sum_{l=1}^{K-1} exp(\beta_{l0} + \beta_l^T x)}$$

- For $Pr(G = k|X = x)$ given above, obviously
  1. Sum up to 1.
  2. A simple calculation shows that the assumptions are satisfied.

Fitting Logistic Regression Models

- Criteria: find parameters that maximize the conditional likelihood of G given X using the training data.
- Denote $p_k(x_i;\theta) = Pr(G = k|X = x_i;\theta)$.
- Given the first input x_1, the posterior probability of its class being g_1 is $Pr(G = g_1|X = x_1)$.
- Since samples in the training data set are independent, the posterior probability for the N samples each having class g_i, $i = 1, 2, \cdots, N$, given their inputs $x_1, x_2, \cdots, x_N$ is:

$$\Pi_{i=1}^{N} Pr(G = g_i|X = x_i).$$

The conditional log-likelihood of the class labels in the training data set is

$$L(\theta) = \sum_{i=1}^{N} \log Pr(G = g_i | X = x_i) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta).$$

Binary Classification

- For binary classification, if $g_i = 1$, denote $y_i = 1$; if $g_i = 2$, denote $y_i = 0$.
- Let $p_1(x;\theta) = p(x;\theta)$, then $p_2(x;\theta) = 1 - p_1(x;\theta) = 1 - p(x;\theta)$.
- Since $K = 2$, the parameters $\theta = \{\beta_{10}, \beta_1\}$. We denote $\beta = (\beta_{10}, \beta_1)^T$.

If $y_i = 1$, i.e., $g_i = 1$,

$$\log p_{g_i}(x;\theta) = \log p_1(x;\theta) = 1 \cdot \log p(x;\theta) = y_i \log p(x;\theta).$$

If $y_i = 0$, i.e., $g_i = 2$,

$$\log p_{g_i}(x;\theta) = \log p_2(x;\theta) = 1 \cdot log(1-p(x;\theta)) = (1-y_i)\log(1-p(x;\theta)).$$

Since either $y_i = 0$ or $1 - y_i = 0$, we have

$$\log p_{g_i}(x;\theta) = y_i \log p(x;\theta) + (1-y_i)\log(1-p(x;\theta)).$$

- The conditional likelihood

$$L(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i; \beta) = \sum_{i=1}^{N} [y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta))]$$

- There are $p + 1$ parameters in $\beta = (\beta_{10}, \beta_1)^T$.
- Assume a column vector form for β,x:

$$\beta = \begin{pmatrix} \beta_{10} \\ \beta_{11} \\ \beta_{12} \\ \vdots \\ \beta_{1p} \end{pmatrix}, x = \begin{pmatrix} 1 \\ x_{.,1} \\ x_{.,2} \\ \vdots \\ x_{.,p} \end{pmatrix}$$

- By the assumption of logistic regression model:

$$p(x;\beta) = Pr(G = 1|X = x) = \frac{exp(\beta^T x)}{1 + exp(\beta^T x)}$$

$$1 - p(x;\beta) = Pr(G = 2|X = x) = \frac{1}{1 + \exp(\beta^T x)}$$

- Substitute the above in $L(\beta)$:

$$L(\beta) = \sum_{i=1}^{N} [y_i \beta^T x_i - \log(1 + e^{\beta^T x_i})]$$

To maximize $L(\beta)$, we set the first order partial derivatives of $L(\beta)$ to zero.

$$\begin{aligned}\frac{\partial L(\beta)}{\partial \beta_{1j}} &= \sum_{i=1}^{N} y_i x_{ij} - \sum_{i=1}^{N} \frac{x_{ij} e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \\ &= \sum_{i=1}^{N} y_i x_{ij} - \sum_{i=1}^{N} x_{ij} p(x; \beta) \\ &= \sum_{i=1}^{N} x_{ij} (y_i - p(x; \beta))\end{aligned}$$

for all $j = 0, 1, \cdots, p$.

- In matrix form, we write

$$\frac{\partial L(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - p(x_i;\beta))$$

- To solve the set of p + 1 nonlinear equations $\frac{\partial L(\beta)}{\partial \beta_{1j}} = 0$, $j = 0, 1, \cdots, p$, use the Newton-Raphson algorithm.
- The Newton-Raphson algorithm requires the second-derivatives or Hessian matrix

$$\frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i;\beta)(1 - p(x_i;\beta)).$$

The element on the j-th row and n-th column is (counting from 0):

$$
\begin{aligned}
\frac{\partial^2 L(\beta)}{\partial \beta_{1j} \partial \beta_{1n}} &= -\sum_{i=1}^{N} \frac{(1+e^{\beta^T x_i})e^{\beta^T x_i} x_{ij} x_{in} - (e^{\beta^T x_i})^2 x_{ij} x_{in}}{(1+e^{\beta^T x_i})^2} \\
&= -\sum_{i=1}^{N} x_{ij} x_{in} p(x_i;\beta) - x_{ij} x_{in} p(x_i;\beta)^2 \\
&= -\sum_{i=1}^{N} x_{ij} x_{in} p(x_i;\beta)(1 - p(x_i;\beta))
\end{aligned}
$$

Starting with β_{old}, a single Newton-Raphson update is

$$\beta_{new} = \beta_{old} - (\frac{\partial^2 L(\beta)}{\partial\beta\partial\beta^T})^{-1}\frac{\partial L(\beta)}{\partial\beta}$$

where the derivatives are evaluated at β_{old}.

The iteration can be expressed compactly in matrix form.

- Let $\mathbf{y}$ be the column vector of y_i.
- Let $\mathbf{X}$ be the $N \times (p+1)$ input matrix.
- Let $\mathbf{p}$ be the N-vector of fitted probabilities with i-th element $p(x_i; \beta_{old})$.
- Let $\mathbf{W}$ be an $N \times N$ diagonal matrix of weights with i-th element $p(x_i; \beta_{old})(1 - p(x_i; \beta_{old}))$.
- Then

$$\frac{\partial L}{\partial \beta} = \mathbf{X}^T(\mathbf{y} - \mathbf{p})$$

$$\frac{\partial L^2}{\partial \beta \partial \beta^T} = -\mathbf{X^T W X}$$

- The Newton-Raphson step is

$$\begin{aligned}\beta_{new} &= \beta_{old} + (\mathbf{X^TWX})^{-1}\mathbf{X^T}(\mathbf{y}-\mathbf{p}) \\ &= (\mathbf{X^TWX})^{-1}\mathbf{X^TW}(\mathbf{X}\beta_{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})) \\ &= (\mathbf{X^TWX})^{-1}\mathbf{X^TWz},\end{aligned}$$

  where $z = \mathbf{X}\beta_{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})$.
- If **z** is viewed as a response and **X** is the input matrix, β_{new} is the solution to a weighted least square problem:

$$\beta_{new} = \arg\min_{\beta} (\mathbf{z} - \mathbf{X}\beta)^T \mathbf{W} (\mathbf{z} - \mathbf{X}\beta)$$

- Recall that linear regression by least square is to solve

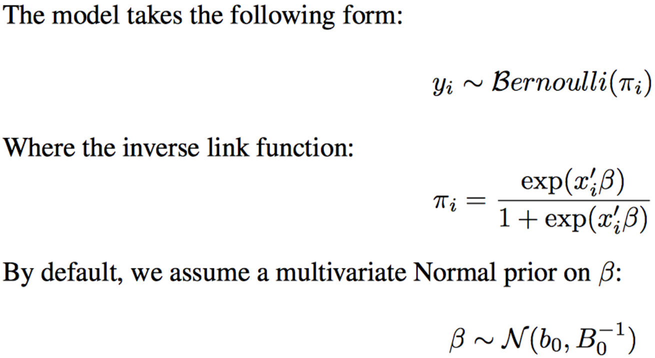
$$\arg\min (\mathbf{z} - \mathbf{X}\beta)^T (\mathbf{z} - \mathbf{X}\beta).$$

- **z** is referred to as the adjusted response.
- The algorithm is referred to as iteratively reweighted least squares or IRLS.

Pseudo Code

```
1. $0 \rightarrow \beta$
2. Compute $y$ by setting its elements to $y_i = 1$ if $g_i = 1$, and $y_i = 0$ if $g_i = 2$, $i = 1, 2, \cdots, N$.
3. Compute $p$ by setting its elements to
       $p(x_i; \beta) = \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}}, i = 1, 2, \cdots, N.$
4. Compute the diagonal matrix $W$. The ith diagonal element is $p(x_i; \beta)(1 - p(x_i; \beta)), i = 1, 2, \cdots, N$.
5. $z \leftarrow X\beta + W^{-1}(y - p)$.
6. $\beta \leftarrow (X^T W X)^{-1} X^T W z$.
7. If the stopping criteria is met, stop; otherwise go back to step 3.
```

Example:Logistic regression: MCMClogit



The model takes the following form:

$$y_i \sim \mathcal{B}ernoulli(\pi_i)$$

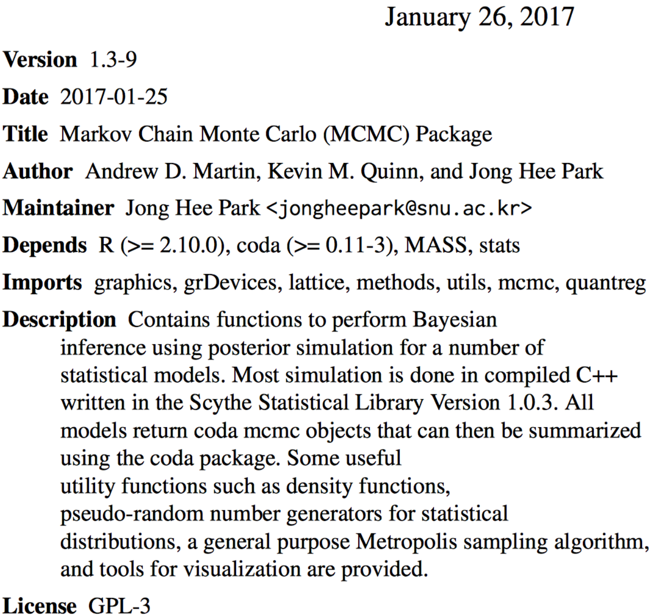
Where the inverse link function:

$$\pi_i = \frac{\exp(x_i'\beta)}{1 + \exp(x_i'\beta)}$$

By default, we assume a multivariate Normal prior on β:

$$\beta \sim \mathcal{N}(b_0, B_0^{-1})$$

R package: MCMCpack



January 26, 2017

Version 1.3-9

Date 2017-01-25

Title Markov Chain Monte Carlo (MCMC) Package

Author Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park

Maintainer Jong Hee Park `<jongheepark@snu.ac.kr>`

Depends R (>= 2.10.0), coda (>= 0.11-3), MASS, stats

Imports graphics, grDevices, lattice, methods, utils, mcmc, quantreg

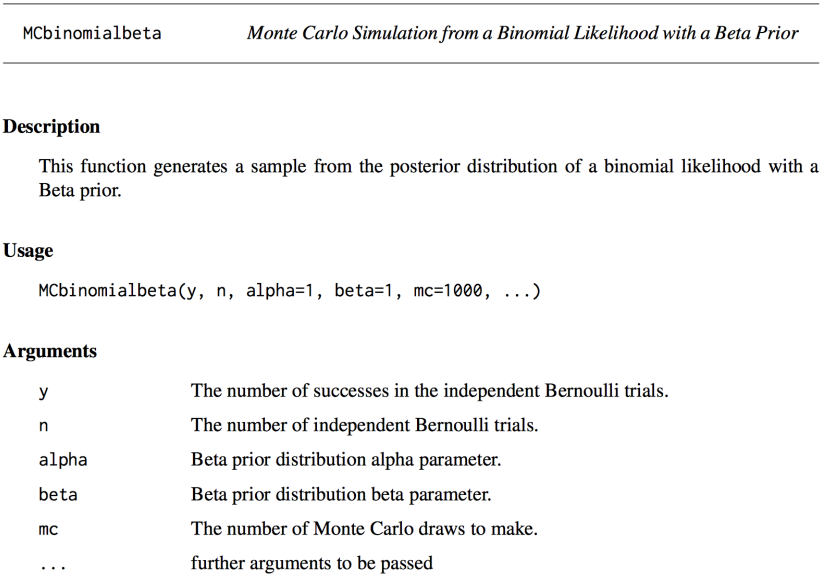
Description Contains functions to perform Bayesian inference using posterior simulation for a number of statistical models. Most simulation is done in compiled C++ written in the Scythe Statistical Library Version 1.0.3. All models return coda mcmc objects that can then be summarized using the coda package. Some useful utility functions such as density functions, pseudo-random number generators for statistical distributions, a general purpose Metropolis sampling algorithm, and tools for visualization are provided.

License GPL-3

Example:Estimating an allele frequency

- Data AA, Aa and aa have frequencies $p \times p, 2 \times p \times (1 - p)$ and $(1 - p) \times (1 - p)$
- Prior: $p \sim U[0, 1]$.
- Likelihood: $(p^2)^{n_{AA}}(2p(1 - p))^{n_{Aa}}((1 - p)^2)^{n_{aa}}$
- Posterior $\propto$ Prior×Likelihood

Example 4: MCbinomialbeta



`MCbinomialbeta` *Monte Carlo Simulation from a Binomial Likelihood with a Beta Prior*

Description

This function generates a sample from the posterior distribution of a binomial likelihood with a Beta prior.

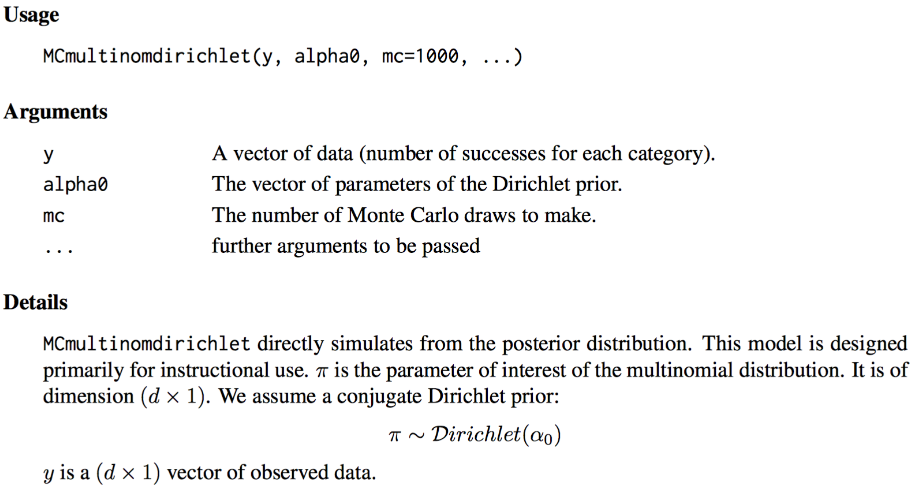
Usage

```
MCbinomialbeta(y, n, alpha=1, beta=1, mc=1000, ...)
```

Arguments

`y`	The number of successes in the independent Bernoulli trials.
`n`	The number of independent Bernoulli trials.
`alpha`	Beta prior distribution alpha parameter.
`beta`	Beta prior distribution beta parameter.
`mc`	The number of Monte Carlo draws to make.
`...`	further arguments to be passed

Example 5: MCmultinomdirichlet



Usage

```
MCmultinomdirichlet(y, alpha0, mc=1000, ...)
```

Arguments

`y`	A vector of data (number of successes for each category).
`alpha0`	The vector of parameters of the Dirichlet prior.
`mc`	The number of Monte Carlo draws to make.
`...`	further arguments to be passed

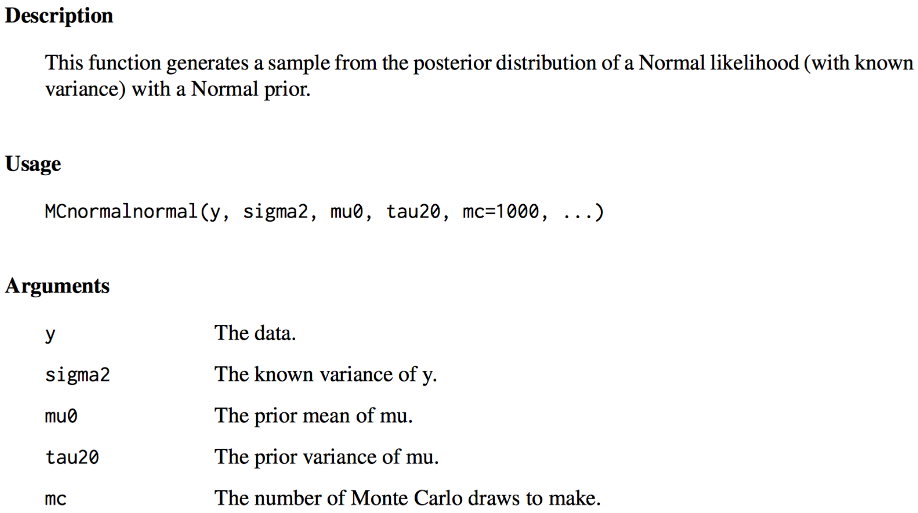
Details

`MCmultinomdirichlet` directly simulates from the posterior distribution. This model is designed primarily for instructional use. π is the parameter of interest of the multinomial distribution. It is of dimension $(d \times 1)$. We assume a conjugate Dirichlet prior:

$$\pi \sim \mathcal{D}irichlet(\alpha_0)$$

y is a $(d \times 1)$ vector of observed data.

Example 6: MCnormalnormal



Description

This function generates a sample from the posterior distribution of a Normal likelihood (with known variance) with a Normal prior.

Usage

```
MCnormalnormal(y, sigma2, mu0, tau20, mc=1000, ...)
```

Arguments

`y`	The data.
`sigma2`	The known variance of y.
`mu0`	The prior mean of mu.
`tau20`	The prior variance of mu.
`mc`	The number of Monte Carlo draws to make.

Example 7: MCpoissongamma

The gamma distribution can be parameterized in terms of a shape parameter $\alpha = k$ and an inverse scale parameter $\beta = 1/\theta$, called a rate parameter. A random variable X that is gamma-distributed with shape α and rate β is denoted

$$X \sim \Gamma(\alpha, \beta) \equiv \mathrm{Gamma}(\alpha, \beta)$$

The corresponding probability density function in the shape-rate parametrization is

$$f(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} \quad \text{for } x > 0 \text{ and } \alpha, \beta > 0,$$

where $\Gamma(\alpha)$ is a complete gamma function.

Example 7: MCpoissongamma

If our data $X_1, \ldots, X_n$ are iid Poisson(λ), then a gamma(α, β) prior on λ is a **conjugate** prior.
Likelihood:

$$L(\lambda|\mathbf{x}) = \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} = \frac{e^{-n\lambda}\lambda^{\sum x_i}}{\prod_{i=1}^{n}(x_i!)}$$

Prior:

$$p(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}, \quad \lambda > 0.$$

$\Rightarrow$ Posterior:

$$\pi(\lambda|\mathbf{x}) \propto \lambda^{\sum x_i+\alpha-1}e^{-(n+\beta)\lambda}, \quad \lambda > 0.$$

$\Rightarrow \pi(\lambda|\mathbf{x})$ is gamma$\left(\sum x_i + \alpha, n + \beta\right)$. **(Conjugate!)**

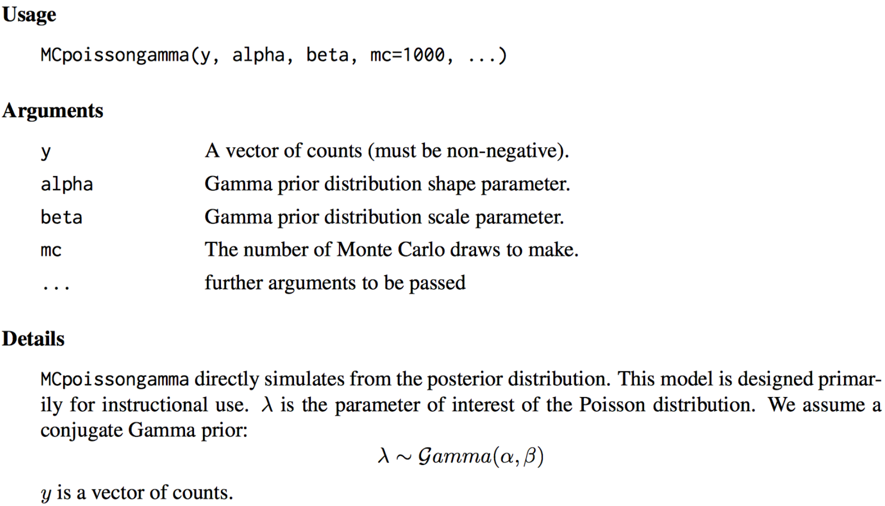
Example 7: MCpoissongamma

The posterior mean is:

$$\begin{aligned}\hat{\lambda}_B &= \frac{\sum x_i + \alpha}{n + \beta} \\ &= \frac{\sum x_i}{n + \beta} + \frac{\alpha}{n + \beta} \\ &= \left[\frac{n}{n+\beta}\right]\left(\frac{\sum x_i}{n}\right) + \left[\frac{\beta}{n+\beta}\right]\left(\frac{\alpha}{\beta}\right)\end{aligned}$$

Again, the data get weighted more heavily as $n \to \infty$.

Example 7: MCpoissongamma



Usage

```
MCpoissongamma(y, alpha, beta, mc=1000, ...)
```

Arguments

`y`	A vector of counts (must be non-negative).
`alpha`	Gamma prior distribution shape parameter.
`beta`	Gamma prior distribution scale parameter.
`mc`	The number of Monte Carlo draws to make.
`...`	further arguments to be passed

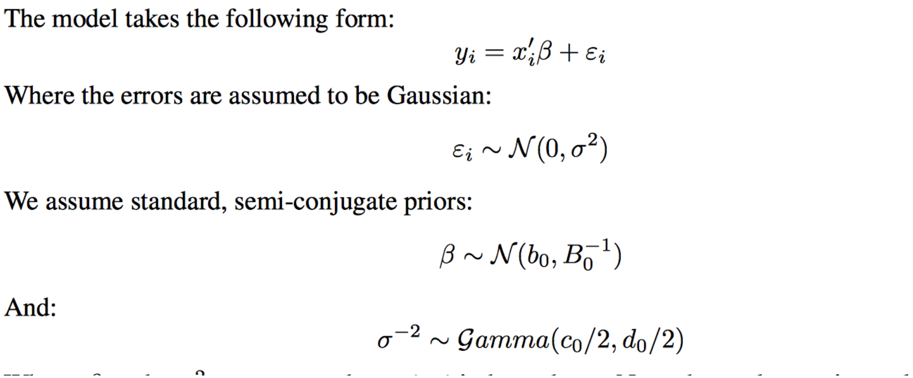
Details

`MCpoissongamma` directly simulates from the posterior distribution. This model is designed primarily for instructional use. λ is the parameter of interest of the Poisson distribution. We assume a conjugate Gamma prior:

$$\lambda \sim \mathcal{G}amma(\alpha, \beta)$$

y is a vector of counts.

Example: Linear regression: MCMCregress



The model takes the following form:

$$y_i = x_i'\beta + \varepsilon_i$$

Where the errors are assumed to be Gaussian:

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

We assume standard, semi-conjugate priors:

$$\beta \sim \mathcal{N}(b_0, B_0^{-1})$$

And:

$$\sigma^{-2} \sim \mathcal{G}amma(c_0/2, d_0/2)$$

Simple Gaussian example: Bayes vs ML

- ML estimate of θ at time n is simply

$$\theta_{ML} = \arg\sup \Pi_{i=1}^{n} f(x_i|\theta) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Posterior of θ at time n is

$$\theta|x_1, \cdots, x_n \sim N(m_n, \sigma_n^2),$$

where

$$\frac{1}{\sigma_n^2} = \frac{1}{\sigma_0^2} + \frac{n}{\sigma^2} \Rightarrow \sigma_n^2 = \frac{\sigma_0^2\sigma^2}{n\sigma_0^2 + \sigma^2} \sim_{n\to\infty} \frac{\sigma^2}{n}$$

$$m_n = \sigma_n^2\left(\frac{\sum_{i=1}^{n} x_i}{\sigma^2} + \frac{m_0}{\sigma_0^2}\right) \sim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Asymptotically in n the prior is washed out by the data and $E[\theta|x_1, \cdots, x_n] = m_n \approx \theta_{ML}$.