Mixture Model in the Analysis of Count Data

Statistical methods in data science

Olga Lazareva

September 2017

1 Introduction

The aim of the project is to perform full Bayesian analysis of the data on the effect of a drug used to treat patients with frequent premature ventricular contractions (PVCs) of the heart. The desirable outcome of the Bayesian analysis is to provide a possible model that explains the data and provide us with measurement of a latent variable which is supposed to give us an information about efficiency of using the drug. The project is organized as follows. Firstly, we will try to understand the data and derive a proper posterior distribution which allow us to make an inference on giving parameters and latent ones. Then we perform a simulation study and try to figure out the way to get accurate estimates. At the end we compare our results with WinBUGS model using DIC and marginal likelihood estimation.
All functions and results are showed in the attached .R file.

1.1 The data

In 1987 Berry presented data on the effect of an antiarrythmic drug for people who suffer frequent premature ventricular contructions (PVCs). In this study the number of PVS was recorded before and after the drug was administered. Here is the data suggested for the analysis:

Table 1: Experiment data

Patient number (i)	PVCs per minute		
	predrug (x_i)	postdrug (y_i)	decrease
1	6	5	1
2	9	2	7
3	17	0	17
4	22	0	22
5	7	2	5
6	5	1	4
7	5	0	5
8	14	0	14
9	9	0	9
10	7	0	7
11	9	13	-4
12	51	0	51

As you can see in *Figure 1*, the postdrug condition is generally better than predrug one, but it's complicated to determine which group of patients was actually cured and which one is experiencing a varying level of response on the drug. Thus, we'd like to measure the probability that a subject was cured or, if not, what kind of improvement we can expect.

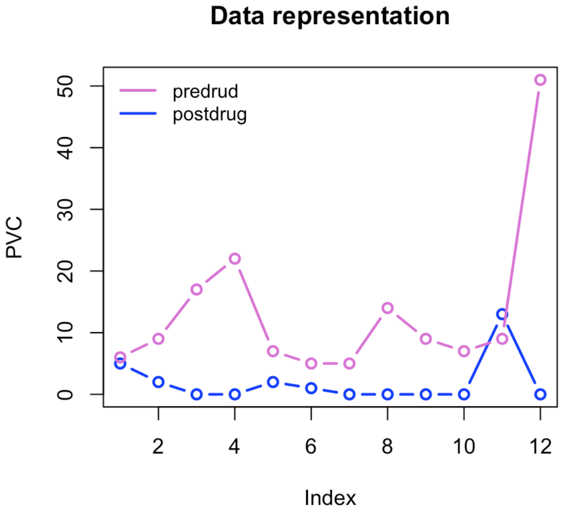

Figure 1: Experiment data

2 A mixture model

2.1 Likelihood

The observations represented as paired data (x_i, y_i), which are the predrug and postdrug counts respectevely, for the i-th patient. We assume that $x_i \sim Poisson(\lambda_i)$ and y_i is independent and $y_i \sim Poisson(\lambda_i\beta)$. Where β represents the expected change in the post-drug PVSs or in other words an expected impact of the drug. Thus, since we have a practical interest in β rather then λ Farewell and Sprott suggested to use conditional distribution of y_i given $t_i = x_i + y_i$. Let's derive it now and see what we can figure out from it:

$$Pr(y_i|t_i = x_i + y_i) = \frac{Pr(y_i, x_i = t_i - y_i)}{Pr(t_i = x_i + y_i)} \tag{1}$$

Let's have a look on numerator and denominator of the equation (1) separately. Numerator:

$$Pr(y_i, x_i = t_i - y_i) =$$

using independence of x_i and y_i:

$$Pr(y_i) \times Pr(x_i = t_i - y_i) = \frac{e^{-\lambda\beta}(\lambda\beta)^{y_i}}{y_i!} \times \frac{e^{-\lambda}\lambda^{(t_i - y_i)}}{(t_i - y_i)!} =$$

$$\frac{e^{-(\lambda\beta+\lambda)}}{t_i!}\binom{t_i}{y_i}\lambda^{t_i - y_i}(\lambda\beta)^{y_i}$$

For the denominator of the expression we might use the property that the sum of two independent Poisson distributions is a Poisson distribution, whose parameter is the sum of the parameters of two independent Poisson (proof is in Theorem 3.2.1 in Introduction to Mathematical Statistics by Hogg et al.). Therefore:

$$Pr(t_i = x_i + y_i) = \frac{e^{-(\lambda\beta+\lambda)}}{t_i!}(\lambda\beta + \lambda)^{t_i}$$

Putting everything back together:

$$Pr(y_i|t_i = x_i + y_i) = \binom{t_i}{y_i}\frac{\lambda^{t_i-y_i}(\lambda\beta)^{y_i}}{(\lambda\beta+\lambda)^{t_i}} =$$

Now we can eliminate λ^{t_i} from numerator and denominator:

$$\binom{t_i}{y_i}\frac{\beta^{y_i}}{(\beta+1)^{t_i}} = \binom{t_i}{y_i}(\frac{\beta}{\beta+1})^{y_i}(\frac{1}{\beta+1})^{t_i-y_i} = \binom{t_i}{y_i}(\frac{\beta}{\beta+1})^{y_i}(1-\frac{\beta}{\beta+1})^{t_i-y_i}$$

Let's denote $p = \frac{\beta}{\beta+1}$ and thus we have $Pr(y_i|t_i = x_i + y_i, p) \sim Bin(t_i, p)$:

$$Pr(y_i|t_i, p) = \binom{t_i}{y_i}p^{y_i}(1-p)^{t_i-y_i} \tag{2}$$

Now, let's consider a latent variable z_i which is responsible for the fact if the patient was cured or not:

$$z_i = \begin{cases} 1, cured \\ 0, otherwise \end{cases}$$

The variable has Bernoulli distribution: $z_i \sim Bern(\theta)$, where θ is a probability of being cured. Note, that if z_i is equal to 1 (the patient is cured) then it requires y_i to be equal to 0 or, in other words, we state that the patient is cured if he doesn't have any PVCs after taking the drug. On the other hand, if z_i is equal to 0 then it doesn't put any limits on y_i: the patient may have improvements in his condition or didn't have any. Thus, using this intuition, we can derive:

$$Pr(y_i|\theta, p) = \theta \cdot I_{[0]}(y_i) + (1-\theta)\binom{t_i}{y_i}p^{y_i}(1-p)^{t_i-y_i} \tag{3}$$

And thus the likelihood is:

$$L(\theta, p) = \prod_{i=1}^{n} Pr(y_i|\theta, p) = \prod_{i=1}^{n}[\theta \cdot I_{[0]}(y_i) + (1-\theta)\binom{t_i}{y_i}p^{y_i}(1-p)^{t_i-y_i}] \tag{4}$$

Using R we can make a 3-d plot which can show us the shape of the MLE for θ and p.

According to the equation 4 and the *Figure 2*, ML estimators for θ and p are equal to: 0.575 and 0.386 respectively:

$$\theta_{MLE} = 0.58, p_{MLE} = 0.386$$

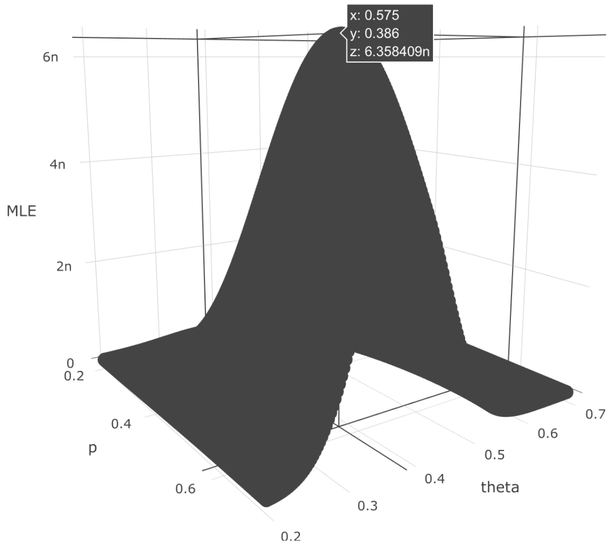

Figure 2: Likelihood function

3 Simulation

3.1 The set-up

Giving the form of the Likelihood we don't have any conjugate prior distributions we could use for θ and p. Thus, the following strategy is suggested.
Firstly, let's use log-odds instead of probabilities. It's quite a common practice in sampling process since it's much convenient to simulate a variable on $(-\infty, \infty)$ (where we can choose any kind of diffuse prior) than on $(0, 1)$. Let's denote:

$$\alpha = log(\frac{p}{1-p}), \delta = log(\frac{\theta}{1-\theta})$$

Now we can choose diffuse priors and perform 'non-informative' inference in order to let the algorithm fully explore posterior distribution.

$$\alpha \sim Norm(0, 10000), \delta \sim Norm(0, 10000)$$

Thus we have everything we need to use Metropolis-Hasting algorithm which will allow us to perform simulation without having conjugate priors.
The simulation was performed with the function **run_Metropolis_MCMC** which takes as an input following parameters:

1. Initial data x,y
2. Start values which were chose as [0,0]

3. thinning parameter t which means that we are going to safe each t-th iteration. For the first run we will set the parameter to 1 and then tune it if it will be necessary

4. Unifa - parameter that was used in the proposal function, which took the previous state of the chain and plus a simulated uniform variable in [-unfa,+unifa]. It was crucial to tune this parameter correct since if the proposal function is narrow compared to the distribution we sample from we get high acceptance rate, but we don't get anywhere. On the other hand, if the proposal function is too wide compared to the distribution we sample from – low acceptance rate, most of the time we stay where we are. In the paper "Weak convergence and optimal scaling of random walk Metropolis algorithms" by A. Gelman et al. it was proven that the optimal acceptance rate is equal to 0.234. Thus unifa-parameter was chosen to get approximately the same rate.

5. Burn-in - a number of first n sample to discard in order to get rid of the bias from initial values. In this case, I've set the initial burn-in to 1000 samples and then we will see how much we actually need.

6. Chain-size - the size of the desirable chain, giving burn-in and thinning. Without burn-in and thinning the chain size is equal to the number of iteration, generally: iterations = chain size*thinning+burn-in, so we should be careful if we don't want a number of iterations to blow up. In this example, I've requested 30000 elements chain.

Also, before going to the results of the simulation, I'd like to describe what estimators I'm going to use in order to describe the results.

1. Mean of the chain;

2. Variance of the chain;

3. 95 % Confidence interval

4. Effective size of the sample which is sort of "exchange rate" between dependent and independent samples. Using this estimator we can evaluate how much information we are actually getting. Thus, if the effective sample size is much smaller than real sample size it means that samples are having very high autocorrelation.

3.2 Simulation and its diagnostic

The chain was generated with the following function **run_metropolis_MCMC(x,y,startvalue = c(0,0), chain_size = 30000,unifa = 1.3,t = 1,burn_in = 1000)**. The results are in close agreement with ML estimators. Here we can say that the probability of being cured is around 60% otherwise we can expect 65% decrees of PVSs.

	α	δ	p	θ	β
Mean	-0.4744	0.3166	0.3856	0.5951	0.6459
Lower 95%	-0.9947	-0.8896	0.2639	0.5656	0.3344
Upper 95%	0.0715	1.5283	0.5106	0.6249	1.0003
Var	0.0756	0.3809	0.0041	0.0002	0.0315
T_eff	5614.9076	2199.9976	-	-	-

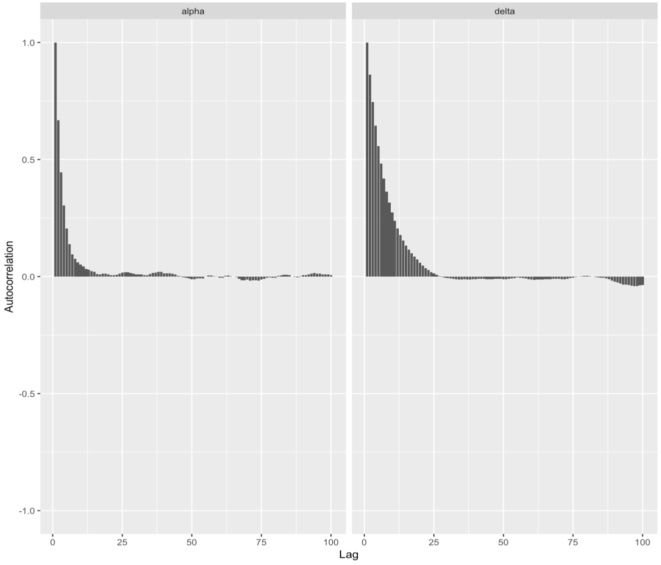

Figure 3: Autocorrelation plot of the first chain with 100 lags

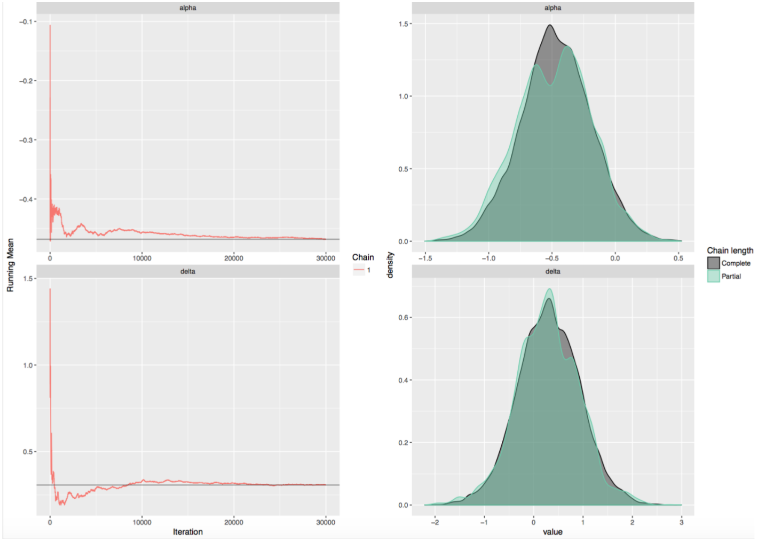

Figure 4: The first chain.Left:running means.Right: density compression

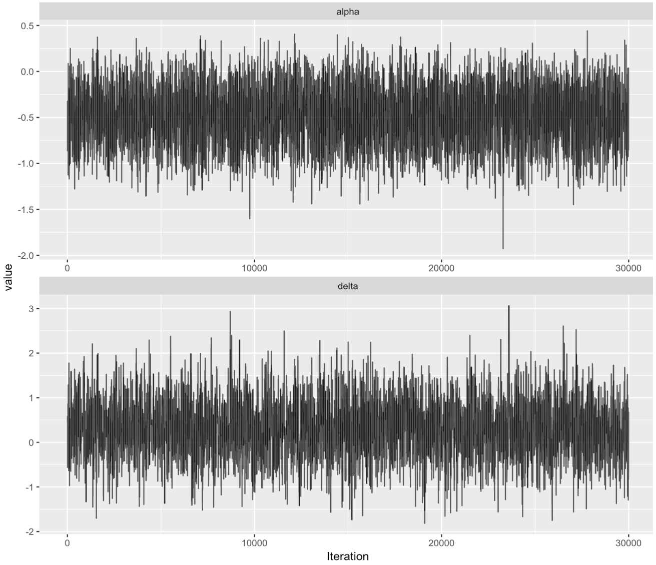

Figure 5: The first chain.Traceplot

At this point I'd like to emphasize that p, θ and β weren't actually simulated, they were calculated from α and δ values using equations described above. Giving this in the further analysis I will refer to results only for α and δ.

We can notice that effective sample size is much smaller than the actual size of the chain and this means that chain has high auto-correlation which makes samples less informative. *Figure 3* confirms this guess. Although it's natural for the Metropolis-Hasting algorithm to have dependent samples giving the way the algorithm works. Since we have significant autocorrelation even after lots of lags it's questionable if thinning will give us any kind of profit in accuracy of the estimation.To get n kept samples in a thinned chain, we needed to generate $n \cdot t$ steps. With such a long chain, the clumpy autocorrelation has probably all been averaged out anyway. So, let's continue to analyze the result and see if we can figure something better. The running means plot (right part of *Figure 4*) shows that we probably have influence of some bias on α since it takes some time to converge. We can check this guess using **Heidelberger and Welch's** convergence diagnostic.

Heidelberger and Welch's convergence diagnostic was performed using **coda** package. The convergence test uses the **Cramer-von-Mises statistic** to test the null hypothesis that the sampled values come from a stationary distribution. The test is successively applied, firstly to the whole chain, then after discarding the first 10%, 20%, ... of the chain until either the null hypothesis is accepted, or 50% of the chain has been discarded. The latter outcome constitutes 'failure' of the stationarity test and indicates that a longer MCMC run is needed. If the stationarity test is passed, the number of iterations to keep and the number to discard are reported.

The result of the diagnostic is such that we need to discard first 3000 iterations in order to obtain a stationary distribution for α. The result of this operation we can see in *Figure 6.*

We can also check **The half-width test** which calculates a 95% confidence interval for the mean. Half the width of this interval is compared with the estimate of the mean. If the ratio between the half-width and the mean is lower than eps = 0.1 , the halfwidth test is passed. Otherwise, the length of the sample is deemed not long enough to estimate the mean with sufficient accuracy.

The halfwidth test was passed for both of the parameters so we can conclude that the estimation of mean is accurate enough.

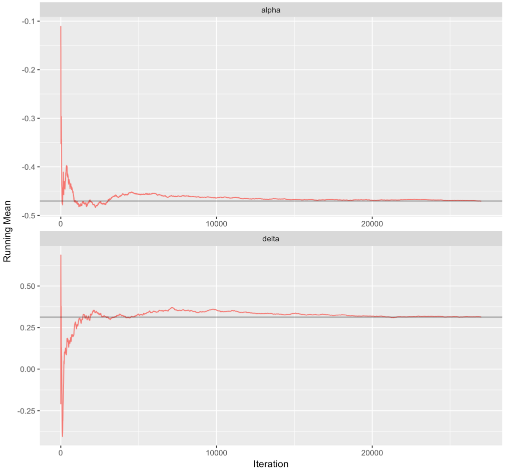

Figure 6: Running means after discarding 3000 iterations

Geweke's convergence diagnostic for Markov chains based on a test for equality of the means of the first and last part of a Markov chain was passed as well
The final result for the current chain is in the following table

	α	δ	p	θ	β
Mean	-0.4759	0.3148	0.3853	0.5951	0.6454
Lower 95%	-0.9992	-0.8613	0.2630	0.5654	0.3344
Upper 95%	0.0715	1.5407	0.5106	0.6249	1.0034
Var	0.0770	0.3782	0.0042	0.0002	0.0320
T_eff	4817.9610	1981.2781	-	-	-

Now we can try to tune thinning parameter and see if the results will improve. It's reasonable to set thinning equal to 50 since we have high autocorrelation up to 50 lags. Giving such a big thinning we will have to make a lot of iterations so we have to know in advance what is the minimum size of the chain. In order to estimate the quantile $q = 0.05$ to within an accuracy of +/- $r = 0.005$ with probability $p = 0.95$ we need a chain which has at least 3746 elements. Therefore, total number of iterations is equal to 3746 × 50 + 1000 (burn-in) which is 188 300. Let's check if it reasonable to do and we will achieve a better result.

New chain was performed with the following parameters run_metropolis_MCMC(x,y,startvalue = c(0,0), chain_size = 3800,unifa = 1.3,t = 50,burn_in = 1000) Having results in the table and at

	α	δ	p	θ	β
Mean	-0.4759	0.3234	0.3852	0.5950	0.6448
Lower 95%	-1.0546	-0.8659	0.2584	0.5642	0.3422
Upper 95%	0.0221	1.5668	0.5055	0.6238	1.0100
Var	0.0746	0.3814	0.0040	0.0002	0.0316
T_eff	3800.0000	3800.0000	-	-	-

Figures 7 and *Figure 8* we can say that even though we've managed to get rid of the autocorrelation

and get a reasonable effective sample size, we can't say that the results became more accurate. Variances of the both estimators are more or less equal and confidence intervals are having the same size.

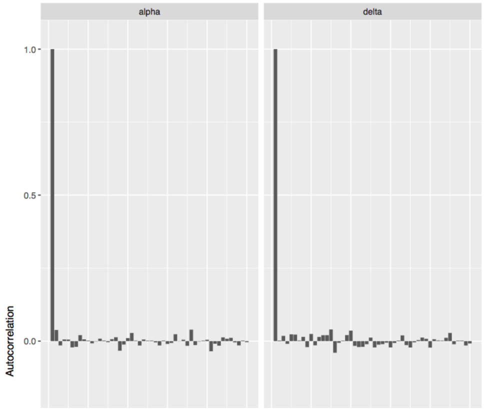

Figure 7: Chain with thinning. Autocorrelation plot

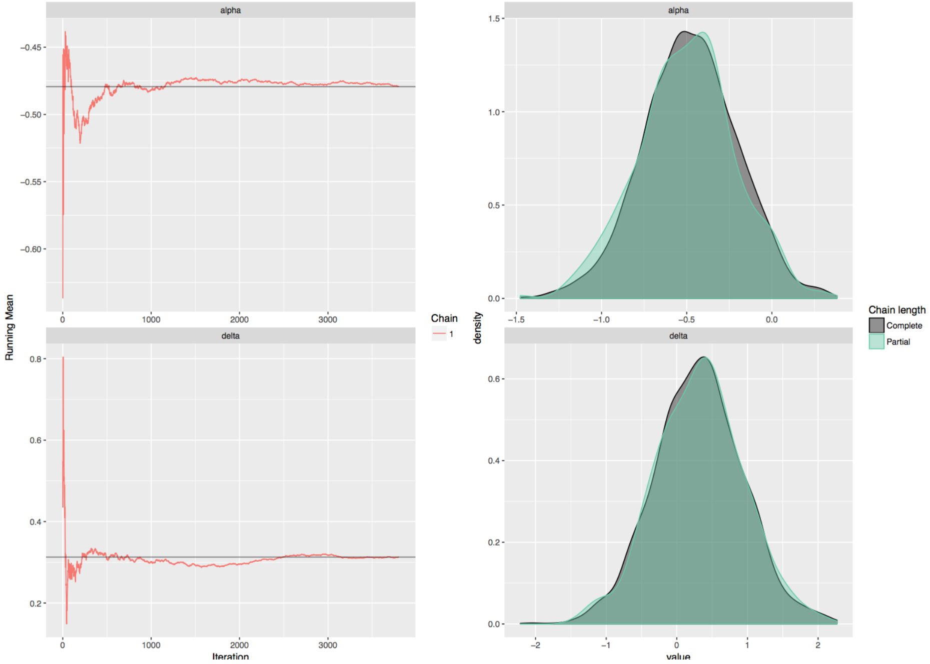

Figure 8: Chain with thinning.Left:running means.Right: density compression

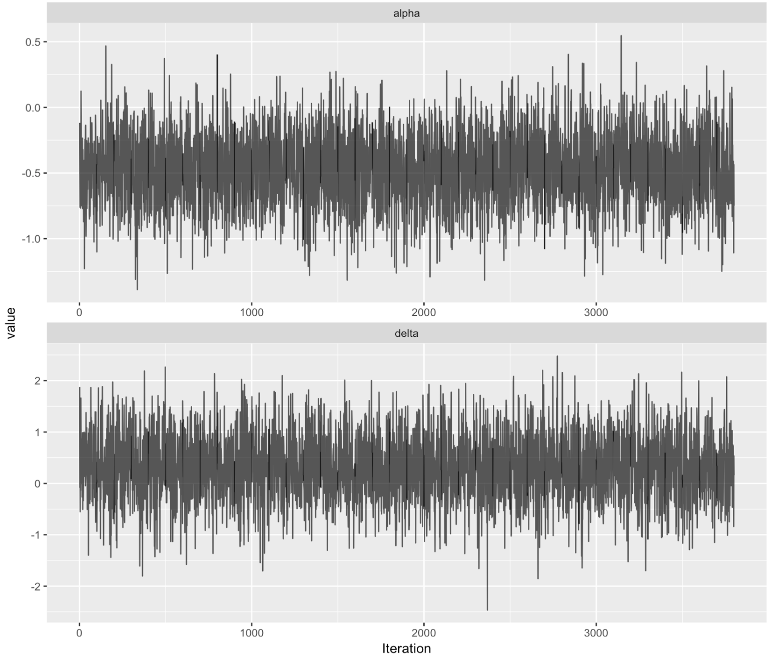

Figure 9: Chain with thinning.Traceplot

4 JAGS model

The following model was suggested:

$$x_i \sim Poisson(\lambda_i)$$

$$y_i \sim Poisson(\lambda_i \beta_i)$$

$$logit(p) = \alpha \sim N(0, 1.0E-4)$$

$$logit(\theta) = \delta \sim N(0, 1.0E-4)$$

$$L(\theta, p) \propto \prod_{i|y_i=0} [\theta + (1-\theta)(1-p)^{t_i}] \cdot \prod_{i|y_i \neq 0} [(1-\theta)p^{y_i}(1-p)^{t_i - y_i}]$$

Initial set-up was [$\delta = 0$, $\alpha = 0$]. There were 20000 iteration with 10000 burn-in and thinning equal to 10. Thus, 1000 iteration were saved.

	α	δ	p	θ	β
Mean	-0.4777	0.2892	0.3850	0.5950	0.6457
Lower 95%	-1.0772	-1.0256	0.2540	0.5632	0.3035
Upper 95%	0.0400	1.4072	0.5100	0.6248	0.9890
Var	0.0817	0.4006	0.0044	0.0003	0.0341
T_eff	1000.0000	1000.0000	-	-	-

From the that above we can see that the chain seems slightly less consistent in comparison with the proposed model: we have larger variance and according to the running means plot, 1000 iterations

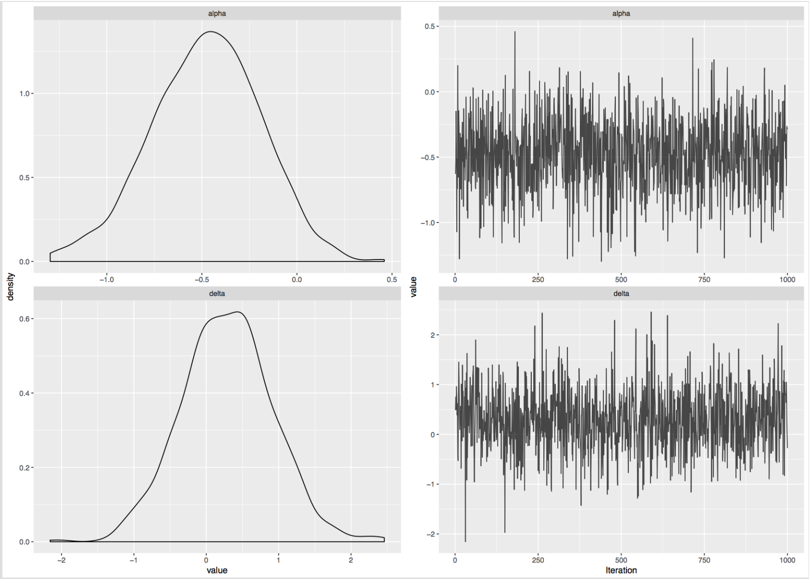

Figure 10: Jags model.Left:density plot.Right: traceplot

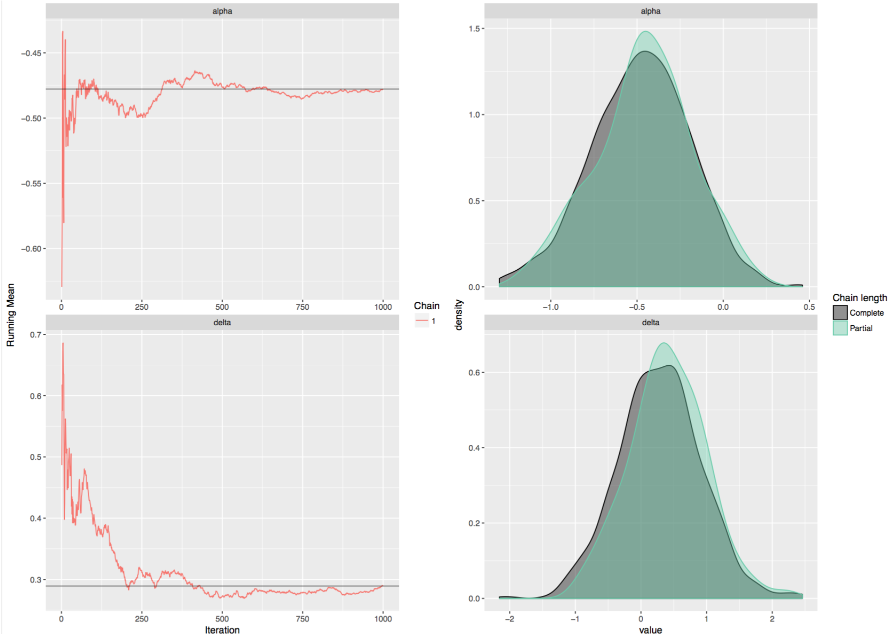

Figure 11: Jags model.Left:running means.Right: density comparison

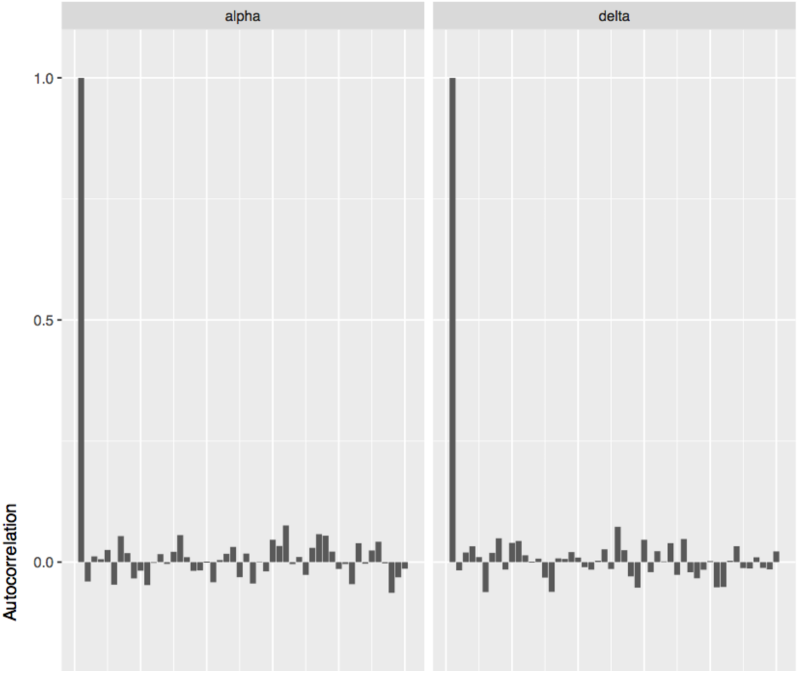

Figure 12: Jags model.Autocorrelation plot

seems not enough to estimate mean of the chain accurate enough.
Heidelberger and Welch's convergence diagnostic confirms the intuition: the current size of the chain doesn't guarantee the desired accuracy for δ.

5 Model selection and comparison

For a model selection, were chosen DIC and Bayesian factor.
Deviance Information Criterion (DIC) which is defined as $D(\theta) = -2log(L(\theta)) + p_D$ where $L(\theta)$ is a likelihood function and p_D is an effective number of parameters. Lower DIC means that the model requires less effective number of parameters to describe the data and hence lower DIC is better.

$$DIC_{proposed} = 41.83523, DIC_{JAGS} = 42.06921$$

Also, I'd like to compare models on the basis of their **marginal likelihood**.
Bayesian model selection proceeds by pairwise comparison of the models in $\{M_l\}$ trough their posterior odds ratio, which for any 2 models M_i M_j is written as:

$$\frac{Pr(M_i|y)}{Pr(M_j, y)} = \frac{Pr(M_i)}{Pr(M_j)} \times \frac{m(y|M_i)}{m(y|M_j)} \tag{5}$$

where

$$m(y|M_i) = \int f(y|M_i, \theta_i)\pi_i(\theta_i|M_i)d\theta_i \tag{6}$$

is the marginal likelihood of M_i and $\frac{Pr(M_i)}{Pr(M_j)}$ is known as prior odds and $\frac{m(y|M_i)}{m(y|M_j)}$ is the Bayes factor.

Evaluating marginal likelihood isn't an easy task since it can't be made directly from the MCMC output and solving the integral analytically isn't an easy option as well. Many approaches were suggested to solve this problem, for instance, Siddhartha Chib and Ivan Jeliazkov suggested their approach in their paper "Marginal likelihood from the Metropolis-Hastings output" which I used for further calculus.
Thus, Bayes Factor between two models is

$$\frac{P(D|M_{prop.})}{P(D|M_{jags})} = 1.001271$$

Giving this result, we can't state that one model is better then another.

6 Results

Putting all the results together:

	Proposed model	Jags model	Farewell and Sprott model
θ	0.5951	0.5950	0.5755
p	0.3853	0.3850	0.3861
β	0.6454	0.6457	0.6289

We can notice that θ estimator in the proposed model and jags model is slightly bigger than in Farewell and Sprott model, but Farewell and Sprott model has quite a large confidence interval [0.30,0.81] while the proposed model confidence interval is [0.56, 0.62] which is much smaller so we can expect it to be more precise.
Generally, all the results are in a close agreement with each other and give us a pretty clear picture of the drug influence. The θ estimate is in accordance with the observed fraction of $\frac{7}{12}$ of patients with zero postdrug counts. Similarly, p for the remaining patients is in agreement with estimated decrease of PVSs. Thus, the proposed model appears to be consistent with the observed data and a subject appears to be either cured with probability around 0.59 or experiencing a 64% improvement.

7 References

Robert V. Hogg, Allen Craig, Joseph W. McKean *Introduction to Mathematical Statistics* 7th Edition.

Steve Brooks, Andrew Gelman, Galin L. Jones and Xiao-Li Mengt *Handbook of Markov Chain Monte Carlo*

V. T. Farewell and D. A. Sprott *The Use of a Mixture Model in the Analysis of Count Data* Biometrics, Vol. 44, No. 4 (Dec., 1988), pp. 1191-1194

William A. Link, Mitchell J. Eaton *On thinning of chains in MCMC* First published on 17 June 2011

A. Gelman, W. R. Gilks, and G. O. Roberts *Weak convergence and optimal scaling of random walk Metropolis algorithms* First published on 17 June 2011