Stat 632 Final Project

F. Jimenez

Abstract

We empirically investigate the impact of centering Bayesian hierarhical models when using Metropolis Hastings and NUTS samplers. A comparison is made with guidelines developed for Gibbs samplers and we conclude with an implementation of a hierarhcal model for the Radon dataset.

1 Introduction

Bayesian hierarhical models contain components which are never observed, but must be infered from the data. The parametrization of said model can drastically impact the performance of MCMC alogirhtms used for inference. This project looks at the importance of parametrization of hierarchical models in light of the results developed by [Papaspiliopoulos et al., 2007]. We begin by briefly summarizing the guidlines developed by [Papaspiliopoulos et al., 2007]. Next we apply these guidelines to the authors' toy problems, but using [Salvatier et al., 2016] and its black-box samplers. Finally, we use these guidelines to more efficiently sample the posteriors for the radon dataset [Gelman, 2006].

2 Parametrizations for Gibbs Sampler

In [Papaspiliopoulos et al., 2007] the authors present the following general model. Assume we have observed data Y, unknown parameters Θ (with prior $P(\Theta)$), and the following probabilistic model $P(Y|\Theta)$. Further assume that this model can be expanded to a hierarchical model by including an unobserved component X in the following way:

$$P(Y|\Theta) = \int P(Y|X,\Theta)P(X|\Theta)d\mu(X). \qquad (1)$$

CODE: github.com/feji3769/Stat632Project

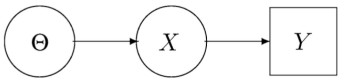
Given this model the authors define a reparametrization scheme as any random pair (X^*, Θ) with joint prior $P(X^*, \Theta)$ and some function h, such that:

$$X = h(X^*, \Theta, Y).$$

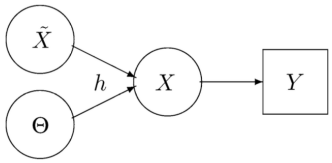
This reparametrization is only intended to improve MCMC performance.

2.1 The Two Parametrizations

The authors present two complementary parametrizations for models in the form of Equation (1): *centered parametrization* (CP) and *hierarchically non-centered parametrization* (NCP). Figure 1 shows the graphical model for both a CP model and a NCP model.



(a) CP graphical model.



(b) NPC graphical model.

Figure 1: The CP model (a) induces strong prior dependence between X and Θ. In the NPC model (b) $\tilde{X}$ and Θ are independent under the prior.

2.2 Criterion for Choosing a Parametrization

To quantify how a parametrization would perform in different settings the authors use the *Bayesian fraction of missing information* [Liu, 1994]:

$$\gamma = \sup_f \left[1 - \frac{E(Var(f(\Theta)|X^*, Y)|Y)}{Var(f(\Theta)|Y)}\right], \qquad (2)$$

The sup is over all square-integrable functions, and γ is the geometric convergence rate of the Gibbs sampler using the (X^*, Θ) parametrization. The value of γ

also corresponds to the auto-correlation within a chain, with large γ resulting in high auto-correlation. Therefore, smaller values of γ mean better convergence. The authors also define $\tau := -1/\log\gamma$, which is proportional to the number of iterations until the chain is close to its stationary distribution.

For any particular hierarchical model the authors define γ_c (τ_c) and γ_{nc} (τ_{nc}) as the *Bayesian fraction of missing information* for the CP and NCP parametrizations respectively.

2.3 General Guidelines

The two parametrizations play complementary roles in hierarchical modeling. In situations where Y is strongly informative of X, the CP model will diminish the posterior dependence between X and Θ. While the NCP model will have high posterior dependence. However, if Y is not strongly informative about X the opposite will be true for both parametrizations.

So given a model one should choose the parametrization which has lower posterior dependence. Since high posterior dependence means high auto-correlation within a chain we can use τ (or *gamma*) to choose between the parametrizations. For a problem where τ_c is smaller than τ_{nc} we would prefer the centered parametrization. If τ_c were larger we'd prefer the non-centered parametrization.

2.4 Focused Guidelines

The advice in the last section is hard to use in practice and to make it more practical the authors study the value of τ for common problems. To do this they examine three subcategories of problems: models whose efficiency depends on the sample size, models whose efficiency depends on the tails the links (think robust models) and models whose efficiency depends on the amount of data being imputed. Under each category they list models and identify which parametrization scheme works best.

3 Toy Examples in PyMC3

In this section we compare CP and NCP parametrizations on some of the examples listed in [Papaspiliopoulos et al., 2007]. We fit the models in PyMC3 [Salvatier et al., 2016] and comment on how our results compare with the guidelines provided in [Papaspiliopoulos et al., 2007]. In PyMC3 we have the option between several MCMC sampling algorithms. We used both NUTS [Hoffman and Gelman, 2011] and Metropolis-Hastings for our toy problems.

3.1 Repeated Measurements

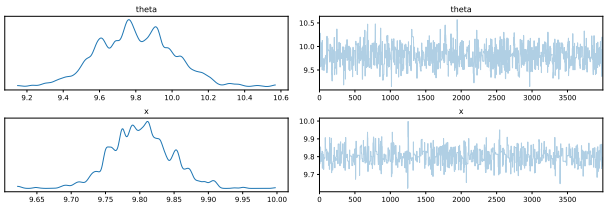
This model is listed under those whose efficiency depends on the sample size, and is expressed as the following:

$$Y_i \sim N(X, \sigma_y^2), i = 1, ..., n$$
$$X = \tilde{X} + \Theta, \tilde{X} \sim N(0, \sigma_x^2)$$

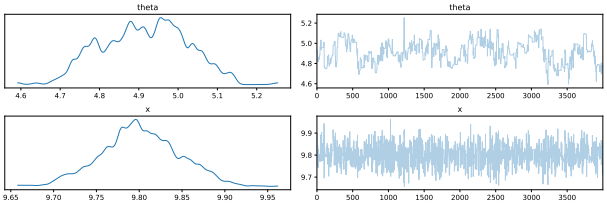
The CP model is $(X*, \Theta)$ and the NCP model is $(\tilde{X}, \Theta)$. In this example $\tau_c = O(1/\log n)$ and $\tau_{nc} = O(n)$, so using a Gibbs sampler we should use the CP parametrization.

For our toy problem we let $n = 5$ and $\Theta = 10$. Then we simulated X from it's prior and then generated each Y_i from the model given. The true value of X was around 9.78.

Figure 2 shows traceplots for Θ using the two parametrizations. When we use Metropolis-Hastings there is a noticeable difference between the mixing of CP and NCP. In fact the effective sample size for Θ using CP was around 580 and for NCP it was 34. Even though we are not using the Gibbs sampler we still have high auto-correlation within our chains. If we use NUTS there is again a noticeable difference in the effective sample size for θ (CP about 3400, NCP about 1000) and NCP fails to capture the true value.



(a) Posterior and traceplots for CP with Metropolis-Hastings. Hastings.



(b) Posterior and traceplots for NCP with Metropolis-Hastings. Hastings.

Figure 2: The CP model (a) traceplots are acceptable and the posterior 95% HDI for Θ contains the truth. The NPC model (b) traceplot for Θ is bad, and the 95% HDI does not contain the truth.

3.2 Cauchy Hidden Markov Model

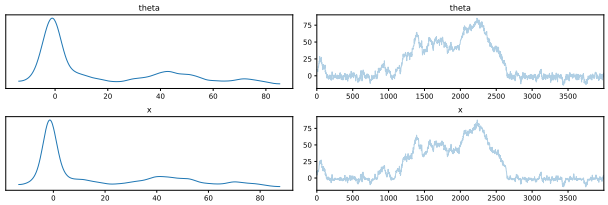
The following Cauchy Hidden Markov Model (HMM) was listed under situations where the efficiency de-

pends upon the tails of the links. It is expressed as:

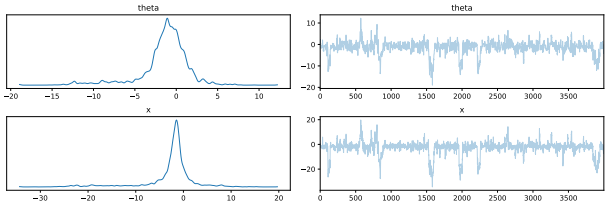
$$Y_i \sim \text{Cauchy}(X_i, \sigma_y^2), \qquad \text{ind. for } i = 1, ..., n,$$
$$X_i = \tilde{X}_i + \Theta, \qquad \tilde{X}_i \sim N(0, \sigma_x^2),$$

Following [Papaspiliopoulos et al., 2007] we let $n = 1$, $\sigma_y = 1, \sigma_x^2 = 5$. We changed the situation slightly by setting $\Theta = 2$ and generating X_1, $\tilde{X}_1$ and Y from the specified model instead of fixing Y. Using the relative values of τ_c and τ_{nc} as our decision rule we ought to prefer NCP over CP in this case.

In Figure 3 we see immediately for both parameters using CP results in a chain that has not reached its stationary distribution after the 1000 burn-in period and isn't stable until almost 3000 post burn-in iterations. NCP doesn't perform perfect as X and Θ have effective sample sizes around 100. That said the 95% HDIs for X and Θ both contain the true values. NCP is clearly an improvement over CP, but running CP for a total of 14000 steps with a burn-in of 5000 produces results similar to NCP in terms of effective sample size and HDIs containing the true values.



(a) Posterior and traceplots for CP with Metropolis Hastings.



(b) Posterior and traceplots for NPC with Metropolis Hastings.

Figure 3: The CP model (a) traceplots are not very good but the 95% HDIs contain the truth. The NPC model (b) results in traceplots that haven't reached their steady state even after the burn-in period.

Using NUTS on this problem does not produce good results for either CP or NCP. The traceplots show the chains have large deviations from their means and the posteriors are very wide as a result. The 95% HDI for Θ under CP is [-19.838, 10.366]. The intervals trivially contain the truth, but don't give us reasonable estimates of uncertainty.

4 Radon Example

The radon dataset contains radon measurements from houses in every county in several states. The data contains, among other things, the following house level data: log-radon level, a binary variable representing whether the measurement came from the basement and what county is the house in. For this analysis we will follow the results presented by [pmy, b]. In that work the author uses data from all the counties in Minnesota and asks if radon measurements in the basement are higher than those on the first floor. To model this behavior the author sets up the following hierarchical model:

$$r_{i,c} = \alpha_c + \beta_c \text{f}_{i,j} + \epsilon_c, \qquad i = 1, ..., n_c; n_c = 1, ..., K,$$
$$\alpha_c \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2), \qquad \beta_c \sim \mathcal{N}(\mu_\beta, \sigma_\beta^2),$$

where K is the number of counties and $f_{i,c}$ is one if the measurement was taken on the first floor. This model assumes that each county, c, has it's own α_c and β_c. Those county means are normally distributed around state-wide means μ_α and μ_β with variance σ_α^2 and σ_β^2. While determining if μ_β is non-zero is of interest in the orginal analysis done by [pmy, a], we are only focused on which parametrization works best for this model. In CP/NCP parlance, should we use the CP or NPC parametrization for α_c and β_c? To answer this question we fit both the CP and NPC parametrizations and compare the results.

After fitting the CP model we can look at the traceplots for a single chain. The traceplots for all the parameters except σ_b were rather good, so we only include σ_b and one set of the parameters we may reparametrize, the β_c. From Figure 4 it's easy to see the chain for σ_b has regions of high auto-correlation. Meaning the chain is struggling to explore a particular region of space.

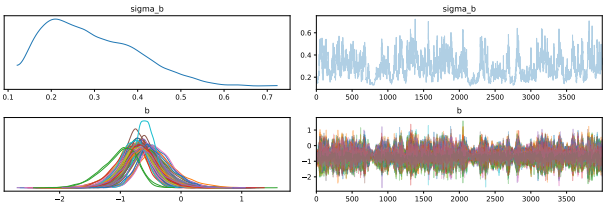


Figure 4: Traceplots and posteriors for σ_b and β_cs using the CP parametrization. The trace plot for σ_b has sequences with high auto-correlation, but all the other parameters have seemingly good traceplots.

In Figure 6 of the joint posterior of σ_b and β_cs (c=75) using the CP parametrization (blue dots) we see there

is a region of this space which isn't being explored. To fix this issue we fit the model using the NPC parametrization. The new traceplots for β_c and σ_b are shown in Figure 5. It looks like we have reduced the auto-correlation within σ_b and improved mixing.

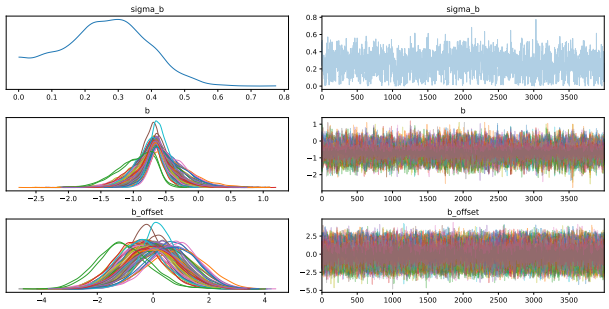


Figure 5: Traceplots and posteriors for σ_b and β_cs using the NCP parametrization. It is clear the auto-correlation within the σ_b chain has decreased.

Next we focus on the joint posterior as before, but using the NPC parametrization. Figure 6 shows the two joint posteriors on top of one another. The circled region was previously unexplored, but is now filled with samples (red dots). From Figure 5 and Figure 6 it is clear that this problem benefits from using a non-centered model as opposed to a centered one.

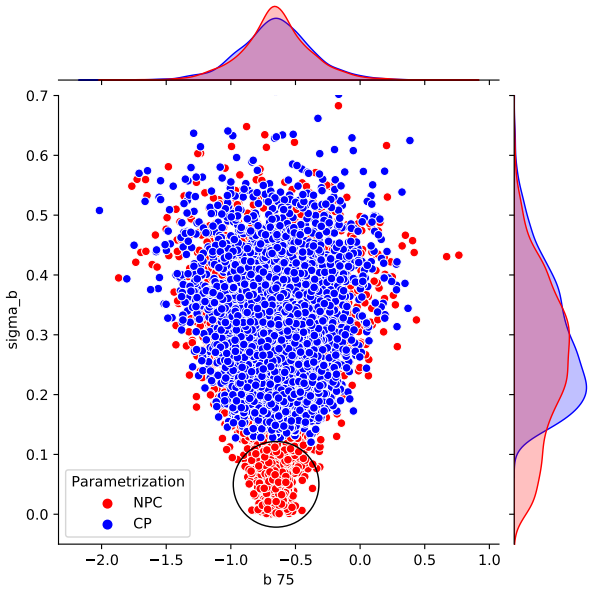


Figure 6: Joint posterior for σ_b (sigma_b) and β_{75} (b 75) using the CP parametrization (blue) and the NPC parametrization (red). The region circled in black is unexplored using CP, but is well sampled under NCP. The marginal distributions of σ_b on the right side of the figure show how the tails of σ_b's posterior have filled in.

5 Conclusion

We presented a summary of the work done by [Papaspiliopoulos et al., 2007] which included guidlines on when to center Bayesian hierarhical models. Then we implemented of some of their examples using Metrpolis Hastings and NUTS samplers and compared our results with their guidelines. For the examples we chose our results were consistent with what was advised.

Using what we learned on toy problems we explored the impact of centering on the Radon dateset [Gelman, 2006, pmy, a, pmy, b] and found the non-centered parametrization greatly improved performance as has been noted before [pmy, b].

6 Discussion

Knowing which parametrization will yield the best results is crucial for conducting a high quality analysis. From our exploration it seems some guidlines extend beyond Gibbs samplers . However, given how simple it is to implement hierarchical models in software it is likely that one can exhaustively search the most likely set of parametrizations successfully. That said general guidelines can speed up the modeling process and therefore have value.

References

[pmy, a] The best of both worlds: Hierarchical linear regression in pymc3. https://twiecki.io/blog/2014/03/17/bayesian-glms-3/. Accessed: 05-05-2020.

[pmy, b] Why hierarchical models are awesome, tricky, and bayesian. https://twiecki.io/blog/2017/02/08/bayesian-hierchical-non-centered/. Accessed: 05-05-2020.

[Gelman, 2006] Gelman, A. (2006). Multilevel (hierarchical) modeling: what it can and cannot do. *Technometrics*, 48(3):432–435.

[Hoffman and Gelman, 2011] Hoffman, M. D. and Gelman, A. (2011). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo.

[Liu, 1994] Liu, J. (1994). Fraction of missing information and convergence rate of data augmentation. In *In Computing Science and Statistics: Proc. 26th Symposiom on the Interface*, page 490– 496, Fairfax Station, VA.

[Papaspiliopoulos et al., 2007] Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). A general

framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73.

[Salvatier et al., 2016] Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in python using PyMC3. *PeerJ Computer Science*, 2:e55.