Computational Statistics : Assignment 4

Raphaël Bernas

December 2024

1 Exercice 1

1.1 Q1.A

Let us note π, the target distribution on $\mathbb{R}^2$, given by

$$(x, y) \mapsto \pi(x, y) \propto \exp\left(-\frac{x^2}{a^2} - y^2 - \frac{1}{4}\left(\frac{x^2}{a^2} - y^2\right)^2\right)$$

where $a > 0$.

Let us consider a Markov transition kernel P defined by

$$P = \frac{1}{2}(P_1 + P_2),$$

where $P_i((x, y), dx' \times dy')$ for $i = 1, 2$ is given by:

$$P_i((x, y), dx' \times dy') = \begin{cases} \mathcal{N}(x' \mid x, \sigma_i^2)\delta(y' - y)\, dx' & \text{if } i = 1, \\ \delta(x' - x)\mathcal{N}(y' \mid y, \sigma_i^2)\, dy' & \text{if } i = 2, \end{cases}$$

where $\mathcal{N}(x' \mid x, \sigma_i^2)$ denotes the Gaussian distribution with mean x and variance σ_i^2, and δ is the Dirac delta function.

The kernel P_i corresponds to a symmetric random walk proposal mechanism that updates only the i-th component while keeping the other component fixed.

The goal of this question is to produce a Markov chain based on distribution π using Metropolis-Hasting algorithm (With kernel P).

You can find all implemented algorithm on this GitHub in the file $TP4.py$:

$$https://github.com/Raphael-Bernas/$$

1.2 Q2.A

For this question let us take $a = 10$, $\sigma_1 = 3$ and $\sigma_2 = 3$. Then we plot the MC obtained and the acceptance rate :

See Figure 1 and Figure 2

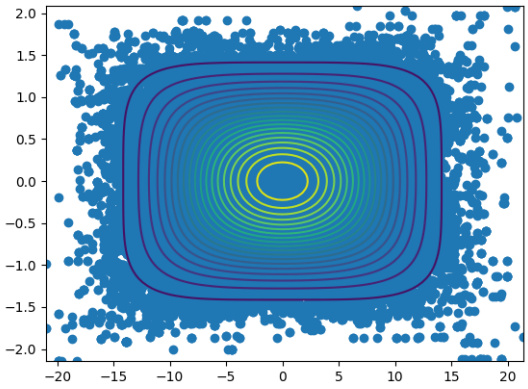

Figure 1: Q2.A. Markov chain generated by Metropolis Hasting for π.

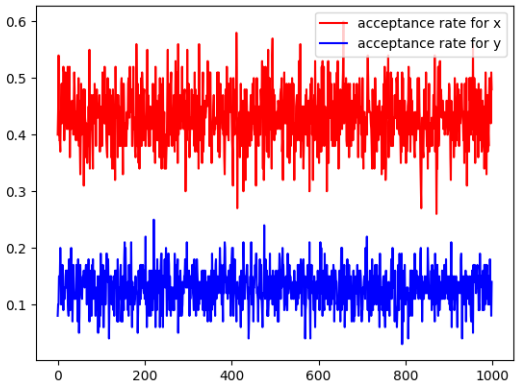

Figure 2: Q2.A. Acceptance rate each for 100 MH step.

Using Figure 1 we observe that this method produce many out-of-distribution point. Mostly around the corner of the distribution, and on the x-axis. This is explained by Figure 2 where it appears that the acceptance rate for x-axis is far above the one for y-axis step.

1.3 Q3.A

- First, to improve this algorithm we could modify the kernel.
- Second, to improve this algorithm we could search for better parameters.

For example, in the first case we could change the kernel used in our Metropolis Hasting algorithm to a new kernel with higher probability to execute a y-step (See Figure 3) :

$$P = \alpha P_1 + (1 - \alpha) P_2 \text{ where } \alpha < \frac{1}{2}$$

And for the second case, choosing parameters with lower values for the x-step variance σ_1 could also be interesting (See Figure 4).

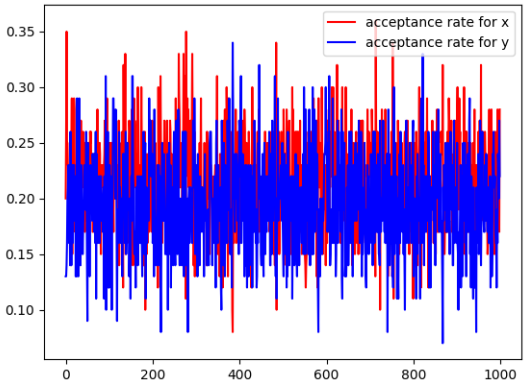

Figure 3: Q3.A. Acceptance rate each for 100 MH step where $\alpha = 0.2 < 0.5$.

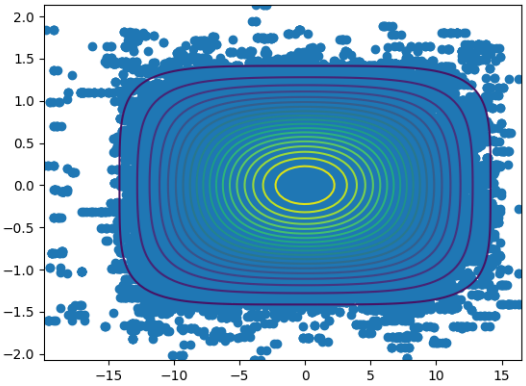

Figure 4: Q3.A. Markov chain generated by Metropolis Hasting for π whith $(\sigma_1, \sigma_2) = (0.5, 3)$.

1.4 Q1.B

Algorithm 1: Adaptive Metropolis-Hastings within Gibbs Sampler

Input: Initial values $x^{(0)} = (x_1^{(0)}, \ldots, x_d^{(0)})$, variances $\sigma_1^2, \ldots, \sigma_d^2$, batch size $B = 50$, target acceptance rate $\rho_{\text{target}} = 0.24$

Initialize:

- Log-variances $\ell_i = \log(\sigma_i)$ for $i = 1, \ldots, d$
- Proposal variances: $\sigma_i = \exp(\ell_i)$
- Iteration counter $k \leftarrow 0$

while *stopping criterion not met* **do**

  for $i = 1$ *to* d **do**

    Proposal: Propose $x_i^* \sim \mathcal{N}(x_i^{(k)}, \sigma_i^2)$

    Acceptance ratio:

$$\alpha(x_i^*, x_i^{(k)}) = \min\left(1, \frac{\pi(x_i^* \mid x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \ldots, x_d^{(k)})}{\pi(x_i^{(k)} \mid x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \ldots, x_d^{(k)})}\right)$$

    Accept or Reject:

$$x_i^{(k+1)} = \begin{cases} x_i^* & \text{with probability } \alpha(x_i^*, x_i^{(k)}) \\ x_i^{(k)} & \text{otherwise.} \end{cases}$$

    Update: Record acceptance for x_i

  if $k \mod B = 0$ *(after every batch of B iterations)* **then**

    for $i = 1$ *to* d **do**

      Compute acceptance rate: ρ_i for the last batch

      if $\rho_i > \rho_{target}$ **then**

        $\ell_i \leftarrow \ell_i + \delta(k)$

      else

        $\ell_i \leftarrow \ell_i - \delta(k)$

      Update proposal variance: $\sigma_i = \exp(\ell_i)$

  Increment: $k = k + 1$

Output: Samples $(x_1^{(k)}, \ldots, x_d^{(k)})$

After computing this Adaptative MH algorithm we obtain those results : See Figure 5

We observe that acceptance rates tend to converge to a similar value around the expected $\rho_{target} = 0.24$. Furthermore, the edge of our modelisation seems smoother which is explainable because the variance can now move. Finally, the autocorrelation shows that we could expect our model to be a good representation of independant variables.

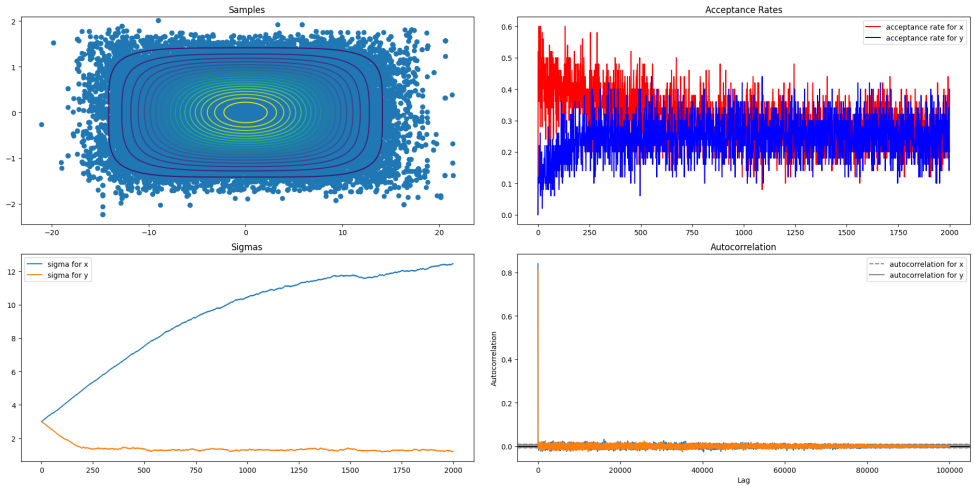

Figure 5: Q1.B. Results for the adaptative method.

But it appears that those results are mitigated in comparison to the possible change we have done on the precedent algorithm.

1.5 Q2.B

Let us denote the following distribution (Banana distribution) :

$$\forall x = (x_1, \dots, x_d) \in \mathbb{R}^d, \quad f_B(x) \propto \exp\left(-\frac{x_1^2}{200} - \frac{1}{2}\left(x_2 + Bx_1^2 - 100B\right)^2 - \frac{1}{2}\left(x_3^2 + \cdots + x_d^2\right)\right).$$

This become our new target distribution. And thus we obtain : See Figure 6

2 Exercice 2

2.1 Q1.A

While using the same adaptative MH algorithm we obtain the following results : See Figure 7. Those results shows that the Markov chain is stuck around one of the gaussian cluster. Furthermore it clearly appears that there is a lot of out-of-distribution point (on the path from the starting point [0,0] to the nearest gaussian cluster).

2.2 Q2.A

Suppose we apply our Metropolis-Hasting algorithm to produce this distribution. Such a method would always lead to the creation of a Markov Chain for which each step is taken on a set distribution around our current point. This is

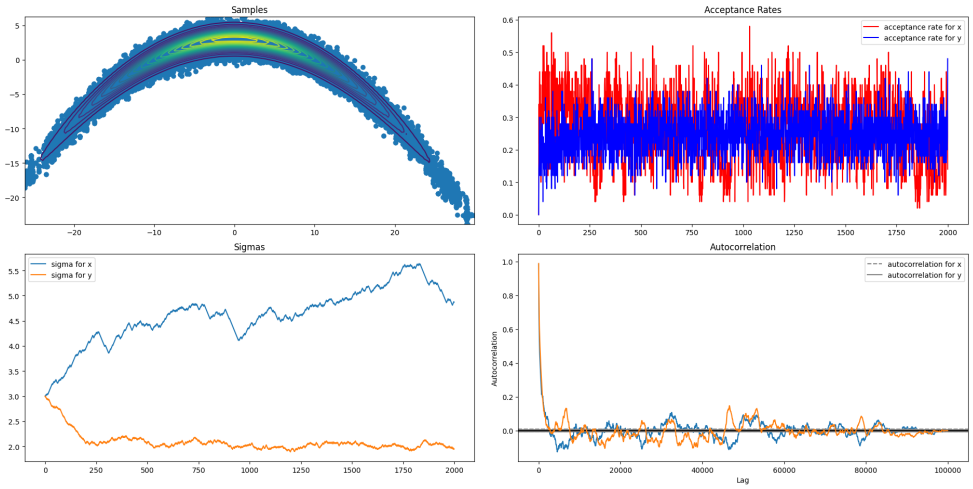

Figure 6: Q2.B. Results for the adaptative method applied to Banana distribution.

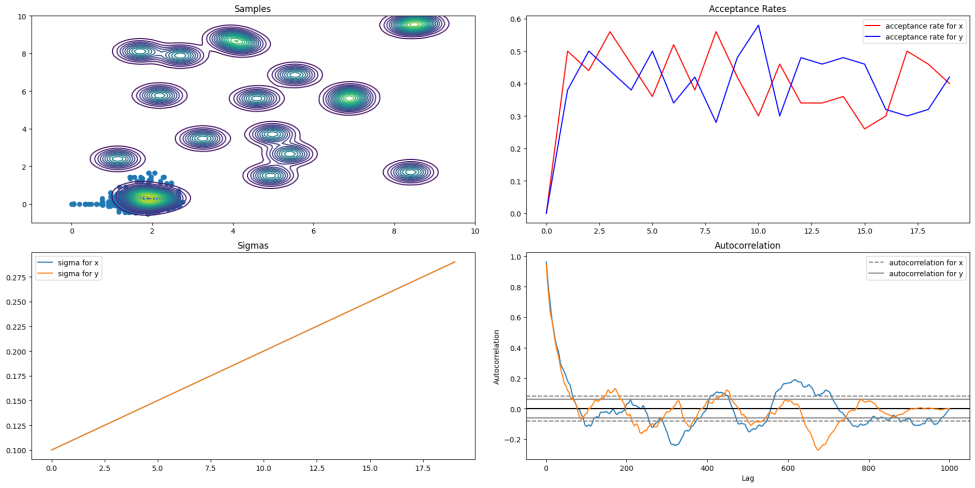

Figure 7: Q1.A. Results for the adaptative method applied to mixed gaussian distribution.

even more true in the case of Gibbs sampler (Random walk).

It would not be an issue in a specific case where the support of the target distribution is a connected space (w.r.t to Lebesgue measure). Indeed, there would be a continuous path to reach every part of the support. Thus, the Markov chain could evolve along those path.

But with our current mixed gaussian distribution, the support is not connected. This means that to pass from one cluster to another the random walk has to generate out-of-distribution point. Therefor, we observe such a limit of Metropolis-Hasting : If there is multiple cluster highly attractive for our distribution, then it could lead to unproper representation of the distribution. Indeed, the Markov chain stays stuck around one cluster. Then a solution could be to increase the variance of our random walk (See Figure 8).

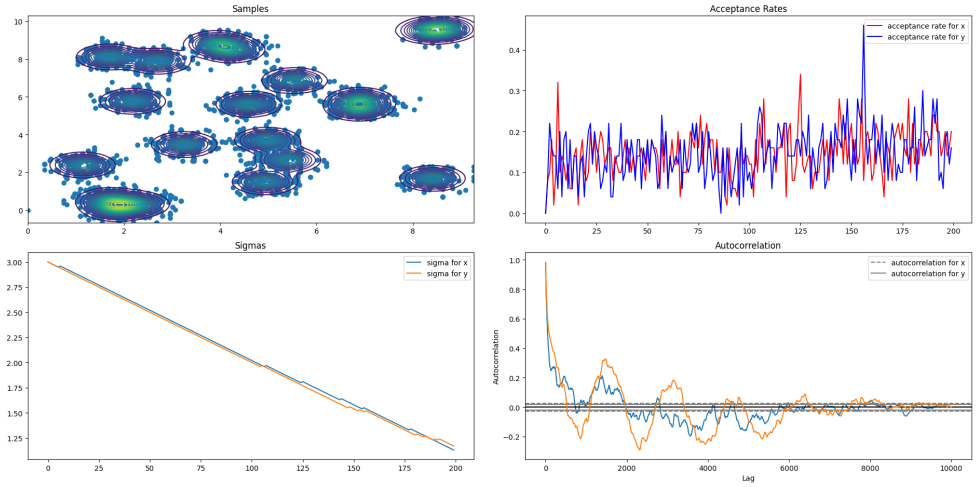

Figure 8: Q2.A. Results for the adaptative method applied to mixed gaussian distribution (with higher variance).

But a new issue arise then, we generate many out-of-distribution point on the path of our Markov chain.

Finally, this problem worsen with the length of the Markov chain. Indeed, the more we sample, the more we pass between distribution. See Figure 9

Thus using Metropolis-Hasting (or adaptative MH) algorithm appears to be unproper in case of unconnected distribution support.

2.3 Q1.B

A solution to overcome those issues is to reduce the "attractivness" of each cluster. To do so, an idea would be to smooth them using a *tempered* version of the distribution:

$$\pi_\beta \propto \pi^\beta, \text{ for } \beta < 1$$

And operate a swap with some probability between those tempered distributions.

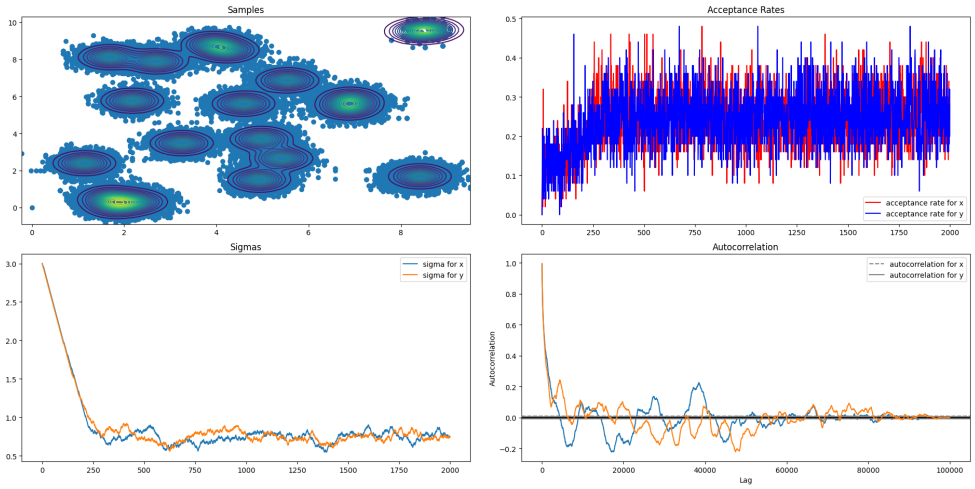

Figure 9: Q2.A. Results for the adaptative method applied to mixed gaussian distribution with more samples.

See Figure 10

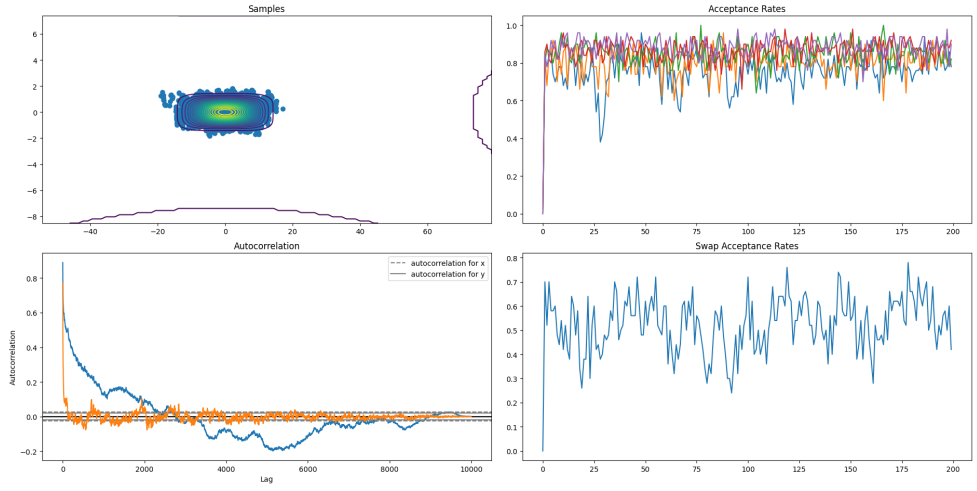

Figure 10: Q1.B. Results for the parallel tempering method applied to the initial distribution.

2.4 Q2.B

Now we compute the mixed gaussian distribution using the parallel tempering method. This give us the following results : See Figure 11.

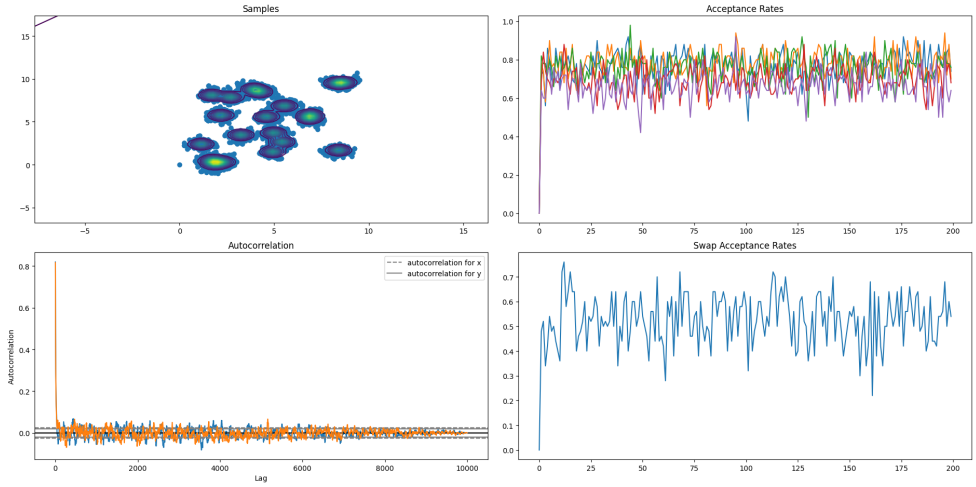

Figure 11: Q2.B. Results for the parallel tempering method applied to the mixed gaussian distribution.

This method clearly changed the capacity of our model to reach unconnected part of the distribution. And as we now have the capacity to "move" greater distance directly, this method is bound to have less out-of-distribution points on the long run. But such an algorithm can have huge impact on the distribution shape. Indeed, with the swapping method some points in a direction of other gaussian cluster (when set in one cluster) become more likely which then reshape a bit the distribution. Cumulating this effect on each gaussian cluster could induce unproper data shape and ultimately mislead someone on the kind of distribution we are sampling.

3 Exercice 3

3.1 Q1.

Inverse Gamma distribution density with positive parameters (a, b) (we note it $\Gamma^{-}_{(a,b)}$):

$$\Gamma^{-}_{(a,b)}(x) \propto \frac{1}{x^{a+1}} \exp\left(-\frac{b}{x}\right) \mathbf{1}_{\mathbb{R}^+}(x),$$

Suppose that we have $Y = \{y_{i,j},\, i \in [|1, N|],\, j \in [|1, k_i|]\}$ and $k := \sum_{i=1}^{N} k_i$ (the total number of observations) such that:

(1) $y_{i,j}$ is a realization of the variable $Y_{i,j}$ where $Y_{i,j} = X_i + \epsilon_{i,j}$;

(2) The random effects $X = \{X_i,\, i \in [|1, N|]\}$ are i.i.d. from a Gaussian $\mathcal{N}(\mu, \sigma^2)$ and independent of the errors $\epsilon = \{\epsilon_{i,j},\, i \in [|1, N|],\, j \in [|1, k_i|]\}$;

(3) The errors ϵ are i.i.d. from the centered Gaussian $\mathcal{N}(0, \tau^2)$.

Bayesian prior distribution for (μ, σ^2, τ^2), the unknown parameters:

$$\pi_{\text{prior}}(\mu, \sigma^2, \tau^2) \propto \frac{1}{\sigma^{2(1+\alpha)}} \exp\left(-\frac{\beta}{\sigma^2}\right) \cdot \frac{1}{\tau^{2(1+\gamma)}} \exp\left(-\frac{\beta}{\tau^2}\right),$$

where α, β, and γ are set hyper-parameters.

Then we want to obtain the density of the posteriori distribution $(X, \mu, \sigma^2, \tau^2)$ up to some constant.

The posterior distribution is proportional to the product of the likelihood and the prior:

$$\pi(X, \mu, \sigma^2, \tau^2 \mid Y) \propto \pi(Y \mid X, \mu, \sigma^2, \tau^2) \cdot \pi(X, \mu, \sigma^2, \tau^2),$$

where:

- The likelihood is:

$$\pi(Y \mid X, \tau^2) = \prod_{i=1}^{N} \prod_{j=1}^{k_i} \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{(y_{i,j} - X_i)^2}{2\tau^2}\right).$$

- The prior is:

$$\pi(X, \mu, \sigma^2, \tau^2) = \pi(X \mid \mu, \sigma^2) \cdot \pi(\mu, \sigma^2, \tau^2),$$

with:

$$\pi(X \mid \mu, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(X_i - \mu)^2}{2\sigma^2}\right),$$

and

$$\pi(\mu, \sigma^2, \tau^2) \propto \frac{1}{\sigma^{2(1+\alpha)}} \exp\left(-\frac{\beta}{\sigma^2}\right) \cdot \frac{1}{\tau^{2(1+\gamma)}} \exp\left(-\frac{\beta}{\tau^2}\right).$$

Combining these, the posterior density is:
$\pi(X, \mu, \sigma^2, \tau^2 \mid Y) \propto \prod_{i=1}^{N} \prod_{j=1}^{k_i} \frac{1}{\sqrt{\tau^2}} \exp\left(-\frac{(y_{i,j} - X_i)^2}{2\tau^2}\right) \cdot \prod_{i=1}^{N} \frac{1}{\sqrt{\sigma^2}} \exp\left(-\frac{(X_i - \mu)^2}{2\sigma^2}\right) \cdot$
$\frac{1}{\sigma^{2(1+\alpha)}} \exp\left(-\frac{\beta}{\sigma^2}\right) \cdot \frac{1}{\tau^{2(1+\gamma)}} \exp\left(-\frac{\beta}{\tau^2}\right).$
which can be rewritten as :

$$\propto \left(\frac{1}{\tau}\right)^{k+2(1+\gamma)} \left(\frac{1}{\sigma}\right)^{N+2(1+\alpha)} \exp\left(-\sum_{i=1}^{N} \left(\left(\sum_{j=1}^{k_i} \frac{(y_{i,j} - X_i)^2}{2\tau^2}\right) + \frac{(X_i - \mu)^2}{2\sigma^2}\right) - \frac{\beta}{\sigma^2} - \frac{\beta}{\tau^2}\right)$$

3.2 Q2.

To implement a Gibbs sampler which would update $(X, \mu, \sigma^2, \tau^2)$ in turns we need before anything else to compute the marginal laws of each parameters :

$$\pi(\sigma^2|X,\mu,\tau^2,Y) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{N}{2}+1+\alpha} \exp\left(-\frac{1}{\sigma^2}\left(\frac{\sum_{i=1}^{N}(X_i-\mu)^2}{2}+\beta\right)\right)$$

$$\sim \Gamma^{-}_{\left(\frac{N}{2}+\alpha\,,\ \frac{\sum_{i=1}^{N}(X_i-\mu)^2}{2}+\beta\right)}(\sigma^2)$$

$$\pi(\tau^2|X,\mu,\sigma^2,Y) \propto \left(\frac{1}{\tau^2}\right)^{\frac{k}{2}+1+\gamma} \exp\left(-\frac{1}{\tau^2}\left(\frac{\sum_{i=1}^{N}\sum_{j=1}^{k_i}(X_i-y_{i,j})^2}{2}+\beta\right)\right)$$

$$\sim \Gamma^{-}_{\left(\frac{k}{2}+\gamma\,,\ \frac{\sum_{i=1}^{N}\sum_{j=1}^{k_i}(X_i-y_{i,j})^2}{2}+\beta\right)}(\tau^2)$$

$$\pi(\mu|X,\sigma^2,\tau^2,Y) \propto \exp\left(\frac{\sum_{i=1}^{N}(\mu-X_i)^2}{2\sigma^2}\right)$$

$$\propto \exp\left(\frac{N\mu^2-2\mu\sum_{i=1}^{N}X_i}{2\sigma^2}\right)$$

$$\propto \exp\left(\frac{(\mu-\frac{1}{N}\sum_{i=1}^{N}X_i)^2}{2\frac{\sigma^2}{N}}\right)$$

$$\sim \mathcal{N}\left(\mu\,\middle|\,\frac{1}{N}\sum_{i=1}^{N}X_i\,,\ \frac{\sigma^2}{N}\right)$$

$$\pi(X|\mu,\sigma^2,\tau^2,Y) \propto \prod_{i=1}^{N}\exp\left(-\frac{(X_i-\mu)^2}{2\sigma^2}-\sum_{j=1}^{k_i}\frac{(X_i-y_{i,j})^2}{2\tau^2}\right)$$

$$\propto \prod_{i=1}^{N}\exp\left(-\left(X_i^2\left(\frac{1}{2\sigma^2}+\frac{k_i}{2\tau^2}\right)-2X_i\left(\frac{\mu}{2\sigma^2}+\frac{\sum_{j=1}^{k_i}y_{i,j}}{2\tau^2}\right)\right)\right)$$

$$\propto \prod_{i=1}^{N}\exp\left(-\frac{\left(X_i-\left(\frac{\mu\tau^2+\sigma^2\sum_{j=1}^{k_i}y_{i,j}}{\tau^2+k_i\sigma^2}\right)\right)^2}{2\left(\frac{\sigma^2\tau^2}{\tau^2+k_i\sigma^2}\right)}\right)$$

$$\sim \prod_{i=1}^{N} \mathcal{N}\left(X_i \,\middle|\, \frac{\mu\tau^2 + \sigma^2 \sum_{j=1}^{k_i} y_{i,j}}{\tau^2 + k_i\sigma^2},\ \frac{\sigma^2\tau^2}{\tau^2 + k_i\sigma^2}\right)$$

Each X_i being independent of each other.

Which give us the following algorithm : See Algorithm 2

Algorithm 2: Gibbs Sampler for Bayesian Hierarchical Model

Input: Observations $Y = \{y_{i,j}, i \in [|1, N|], j \in [|1, k_i|]\}$, hyperparameters α, β, γ, initial values $(\sigma^2, \tau^2, \mu, X)$, number of iterations T

Initialize:

- Initial values: $\sigma^2 = \sigma_0^2$, $\tau^2 = \tau_0^2$, $\mu = \mu_0$, $X = \{X_1^{(0)}, \ldots, X_N^{(0)}\}$
- Iteration counter: $t \leftarrow 0$

while *stopping criterion not met* $(t < T)$ **do**

Update σ^2

Sample:

$$\sigma^2 \sim \Gamma^{-}_{\left(\frac{N}{2}+\alpha, \frac{1}{2}\sum_{i=1}^{N}(X_i^{(t)}-\mu^{(t)})^2+\beta\right)}$$

Update τ^2

Sample:

$$\tau^2 \sim \Gamma^{-}_{\left(\frac{k}{2}+\gamma, \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{k_i}(y_{i,j}-X_i^{(t)})^2+\beta\right)}$$

Update μ

Sample:

$$\mu \sim \mathcal{N}\left(\frac{\sum_{i=1}^{N} X_i^{(t)}}{N}, \frac{\sigma^2}{N}\right)$$

Update $X = \{X_1, \ldots, X_N\}$

For $i = 1, \ldots, N$, sample:

$$X_i \sim \mathcal{N}\left(\frac{\frac{1}{\tau^2}\sum_{j=1}^{k_i} y_{i,j} + \frac{1}{\sigma^2}\mu}{\frac{k_i}{\tau^2} + \frac{1}{\sigma^2}}, \frac{1}{\frac{k_i}{\tau^2} + \frac{1}{\sigma^2}}\right)$$

Increment: $t \leftarrow t + 1$

Output: Posterior samples $(\sigma^2, \tau^2, \mu, X)$

3.3 Q3.

Using Q1 that we can write :

$$\pi(X,\mu \mid Y,\sigma^2,\tau^2) \propto \exp\left(-\frac{1}{2\tau^2}\sum_{i=1}^{N}\sum_{j=1}^{k_i}(y_{i,j}-X_i)^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(X_i-\mu)^2\right)$$

$$\propto \exp\left(-\sum_{i=1}^{N}\left(\frac{1}{2\tau^2}k_i X_i^2 - \frac{2}{2\tau^2}X_i\sum_{j=1}^{k_i}y_{i,j} + \frac{1}{2\sigma^2}X_i^2 - \frac{2}{2\sigma^2}X_i\mu + \frac{1}{2\sigma^2}\mu^2\right)\right)$$

$$\propto \exp\left(-\frac{1}{2}\sum_{i=1}^{N}\left((\frac{k_i}{\tau^2}+\frac{1}{\sigma^2})X_i^2 - \frac{2}{\tau^2}X_i\sum_{j=1}^{k_i}y_{i,j} - \frac{2}{\sigma^2}X_i\mu + \frac{1}{\sigma^2}\mu^2\right)\right)$$

$$\sim \mathcal{N}\left(\Sigma\begin{pmatrix}\frac{\sum_{j=1}^{k_1}y_{i,j}}{\tau^2}\\ \vdots \\ \frac{\sum_{j=1}^{k_N}y_{i,j}}{\tau^2}\\ 0\end{pmatrix},\ \Sigma\right)$$

Where $\Sigma^{-1} = \begin{pmatrix}(\frac{k_i}{\tau^2}+\frac{1}{\sigma^2}) & 0 & 0 & -\frac{1}{\sigma^2}\\ 0 & \ddots & 0 & \vdots\\ 0 & 0 & (\frac{k_i}{\tau^2}+\frac{1}{\sigma^2}) & -\frac{1}{\sigma^2}\\ -\frac{1}{\sigma^2} & \cdots & -\frac{1}{\sigma^2} & \frac{N}{\sigma^2}\end{pmatrix}$

We replace in the precedent Gibbs algorithm the part where we compute X and μ by a computation of (X,μ).

3.4 Q4.

Comparison of Gibbs and Block-Gibbs :
Classic Gibbs sampler has many advantage that should not be underestimated. Indeed, such a method is simpler to implement than Block-Gibbs and furthermore needs potentially less complex law to sample from. Which could be a huge win if we where to sample law using MH algorithm or any computationally expensive algorithm. As we have explicit law in our case, this is not much of an improvement.

However, Block-Gibbs in our peculiar[1] case will be faster with less step. More generally, such a method is bound to be more accurate on the parameter distribution because there are fewer approximation errors induced by the fact that we compute step-by-step.

[1] In the case of explicit law for a block.

Finally, in the case of highly correlated variable, the Gibbs sampler will require more steps to properly converge than Block-Gibbs. Indeed, if we set a block with our highly correlated variables together, then Block-Gibbs can accurately sample from the correct distribution without inducing a small bias[2].

3.5 Q5.

We apply our two algorihtm on a syntheticly generated dataset Y :

See Figure $12, \cdots, 16$

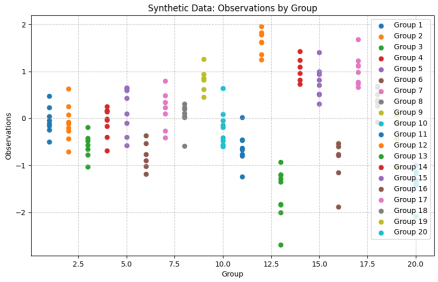

Figure 12: Q5. Synthetic dataset.

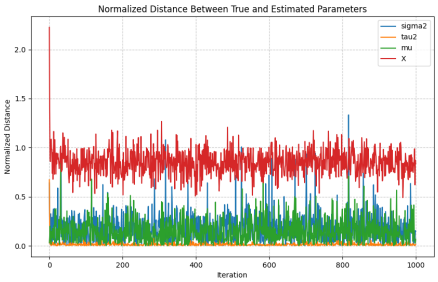

Figure 13: Q5. Gibbs sampler : Square distances between true and estimated parameters.

It appears that the Block-Gibbs sampler did reduce the variance on its "block" parameters (X, μ) but this is not as much marked as we could have expected it. It highly possible that such an example was not enough to be certain.

[2]Which will ultimately disapear after each bayesian update for the classic Gibbs sampler too

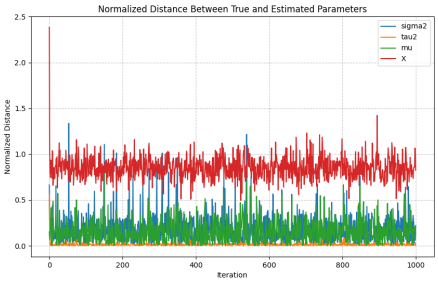

Figure 14: Q5. Block-Gibbs sampler : Square distances between true and estimated parameters.

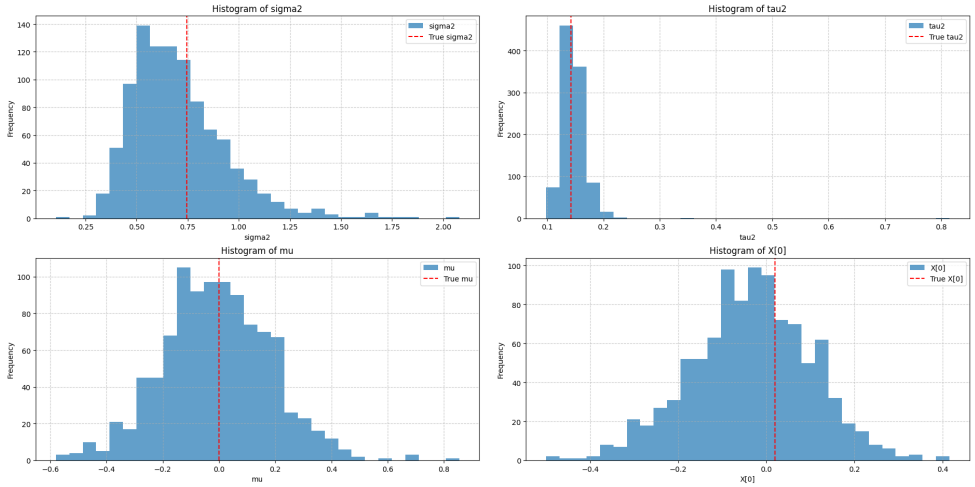

Figure 15: Q5. Gibbs sampler : Histogram of estimated parameters.

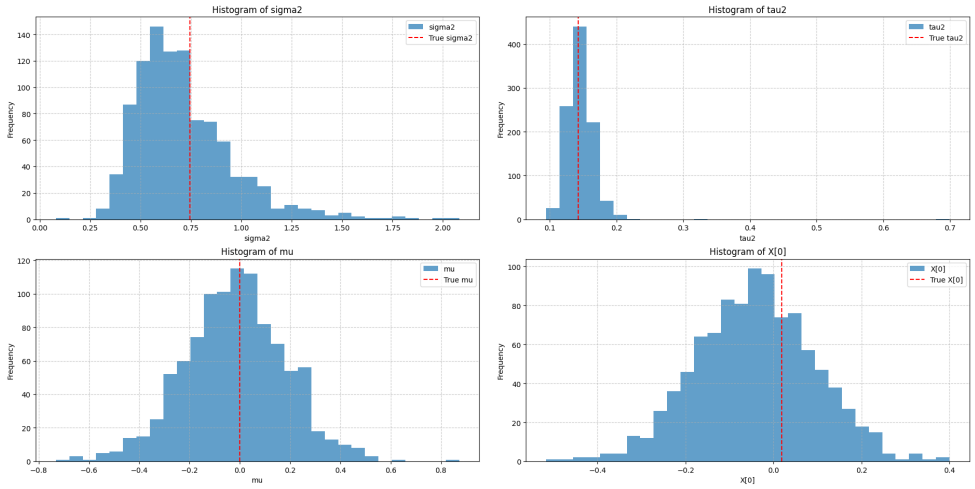

Figure 16: Q5. Block-Gibbs sampler : Histogram of estimated parameters.