Cómputo Científico
Reporte técnico del proyecto final

Joel Chacón Castillo

Guanajuato, México

Los métodos de Metrópolis-Hastings son extremadamente utilizados en inferencia estadística para generar muestrar de distribuciones complicadas. Generar muestras de un modelo con un número de dimensiones excesivo suele causar problemas en los métodos de Metrópolis-Hastings (entre otros) ya que –dependiendo del problema– la velocidad de convergencia puede estar sériamente afectada. Una estrategia para mejorar el rendimiento de los métodos MCMC es considerando mecanismos adaptativos. Los métodos adaptativos MCMC (Markov Chain Monte-Carlo) están generados con el propósito de automáticamente *aprender* mejors valores de la cadena.

En este proyecto se implementan algunos métodos adaptativos propuestos por Gareth O. et al [1].

Es bien conocido que los algoritmos adaptativos MCMC no siempre preservan *estacionariedad* de la distribución objetivo $\pi(.)$. Sin embargo, pueden tener un grado de convergencia si las adaptaciones son realizadas en tiempos de regeneración o bajo ciertas condiciones técnicas acerca del procedimiento de adaptación(Haario, Saksman, and Tamminen ***).

Particularmente Roberts y Rosenthal (2005) probaron ergodicidad bajo ciertas condiciones donde no se requiere que los parámetros adaptativos convergan. Para establecer este resultado, se supone que el algoritmo actualiza el estado X_n al estado X_{n+1} utilizando el kernel P_{Γ_n}, donde cada kernel fijo P_ρ tiene una distribución estacionaria $\pi(.)$, pero donde Γ_n son ínidices aleatorios escogidos iterativamente de una colección Y basada en la salida de un algoritmo anterior.

Considerando $||...||$ como la distancia de varianza total, χ el estado de espacios, y $M_\epsilon(x,\gamma) = inf\{n \geq 1 : ||P_\gamma^n(x,.) - \pi(.)|| \leq \epsilon\}$ es el tiempo de convergencia del kernel P_γ cuando comienza en el etado $x \in \chi$. Entonces el **teorema 13** de Roberts and Rosenthal (2005), garantiza convergencia asintótica $lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} g(X_i) = \pi(g)$ para todo límite $g : \chi \to \Re$ (ley débil de los números grandes), esto sólo si se asume la condición *Diminishing Adaptation*:

$$lim_{n\to\infty} sum_{x\in\chi} sup||P_{\Gamma_{n+1}(x,.)-P_{\Gamma_n}(x,.)}|| = 0 \tag{1}$$

y además considerando la condición *Bounded Convergence*

$$\{M_\epsilon(X_n, \Gamma_n)\}_{n=0}^{\infty} \quad \epsilon > 0 \tag{2}$$

Se probó que la ecuación (1) se satisface si $\chi \times Y$ es finito, o si es una topología compacta en el cual puede pasar que los kernels de transición P_γ o que los kernels propuestos Q_γ tengan densidades conjuntas continuas. En realidad en *Diminishing Adaptation condition* se establece que dos kernels de transición sucesivos son similares y *Boundary Convergence condition* asegura la ergodicidad de los kernels de transición.

1. Adaptive Metropolis (AM)

En esta sección se considera la versión adaptiativa propuesta por Haario, et al [2], donde se considera una distribución objetivo $\pi(.)$. En particular en el trabajo se implementa el algoritmo de Metropolis con ls siguiente distribución propuesta:

$$Q_n(x,.) = \begin{cases} N(x,(0{,}1)^2 I_d/d), & \text{if } n \leq 2d \\ (1-\beta)N(x,(2{,}38)^2\Sigma_n/d) + \beta N(x,(0{,}1)^2 I_d/d), & \text{otherwise} \end{cases} \tag{3}$$

donde Σ_n es la estimación empírica hasta el momento de la estructura de covarianza de la distribución objetivo, y β es una constante prqueña (se consideró $\beta = 0{,}05$). La motivación de esta propuesta está basado en ya se ha probado que $N(x,(2{,}38^2)\Sigma/d)$ es óptimo para un contexto de alta dimensionalidad (Roberts and Rosenthal - 2001). Además se agrega una mezcla con $N(x,(0{,}1)^2 I_d/d)$ para propocionar una medición *segura* y evitar que el algoritmo es estanque con valores problemáticos de Σ_n. Restringiendo que $\beta > 0$ se satisface *Boundary Conditions* al menos para una familia larga de densidades objetivo. Por lo tanto este algoritmo converge a una distribución objetivo. Para probar este algoritmo se diseña la distribución objetivo $\pi(.) = N(0, MM^t)$, donde M es una matriz $d \times d$ dimensional generada de forma aleatoria $\{M_{ij}\}_{i,j=1} \sim N(0,1)$. Esto asegura que la matriz de covarianza sea altamente errática. Por lo tanto realizar muestras de $\pi(.)$ es un reto significativo en dimensiones altas. Para monitorear el éxito de esta implementación se utiliza un factor de suboptimalidad. En particular Roberts et al (2001) probó que es óptimo tomar $\Sigma_p = \Sigma$ y para otro Σ_p la razón de mezcla será menor que el siguiente factor de suboptimalidad

$$b = d\frac{\sum_{i_1}^{d} \lambda_i^{-}2}{(\sum_{i_1}^{d} \lambda_i^{-1)2}} \tag{4}$$

donde $\{\lambda_i\}$ son los valores propios de la matriz $\Sigma_p^{\frac{1}{2}}\Sigma^{-\frac{1}{2}}$. Por lo regular se tiene $b > 1$ y entre más cercano sea b a 1 la estimación de la matriz Σ será mejor. Se implementó este algoritmo y se ejecutó con dimension 100 y $1'000'000$ iteraciones, en la figura 1 se puede observar la convergencia de factor de suboptimalidad (parte izquierda) y el desplazamiento de la primera coordenada (parte derecha). Para confirmar que el algoritmo funciona correctamente se probó con dos dimensiones. En la figura 2 se ilustra el factor de de suboptimalidad (parte superior izquierda), la primer coordenada (parte superior derecha) y el mapa de contonro (parte inferior izquierda). Particularmente se observa que el factor de suboptimzalidad no converge a la unidad, posiblemente esto se debe al sesgo que existe en la estimación de la matriz de covarianza. Por lo tanto se concluye que el algoritmo AM aprende acerca de la matriz de covarianza de la distribución objetivo.

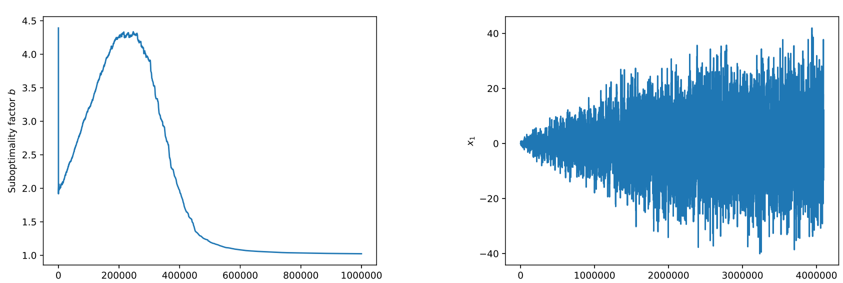

Figura 1: Factor suboptimo (parte derecha) y la primer coordenada del algoritmo considerando dimensión 100 (derecha).

.

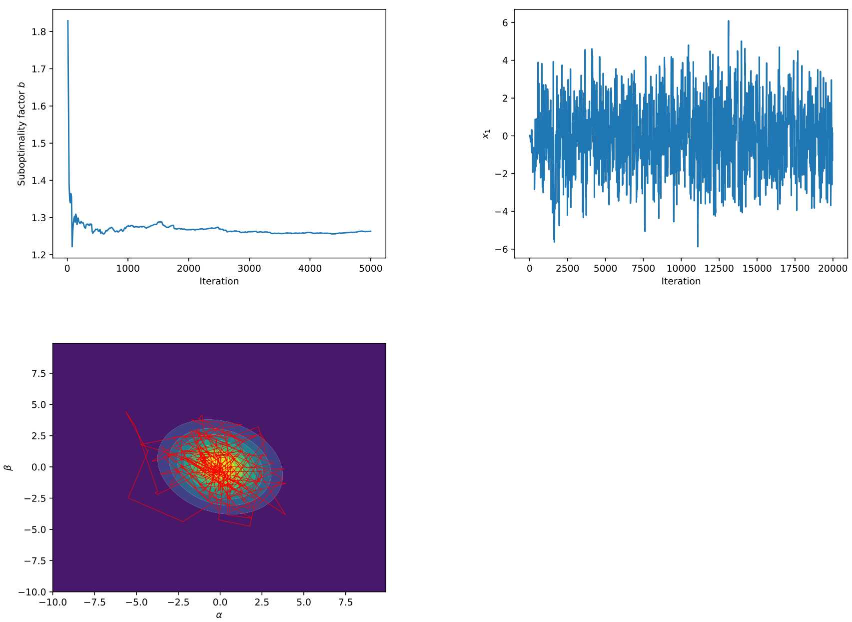

Figura 2: Factor suboptimo (parte derecha) y la primer coordenada del algoritmo considerando dimensión 2 (derecha). Se probaron 1000 iteraciones

.

2. An Irregular Shaped Example

El algoritmo AM puede proporcionar buenos resultados en densidades cuyos contornos con aproximadamente elípticos. En estos ejemplos la covarianza global proporciona una buena medición de la dependencia en todas las partes del espacio de estados. Este algoritmo se implementó con una distribución *banana-shaped* propuesta por Haario, Saksman et al. (1999) con la densidad $f_B = f_d \circ \Phi_B$, donde f_d es una función de densidad d-dimensional de una distribución $N(\mathbf{0}, diag(100, 1, ..., 1))$ y donde $\Phi_B(x_1, ..., x_d) = (x_1, x_2 + Bx_1^2 - 100B, x_3, ..., x_d)^2)$ con $B > 0$ (constante de bananacidad). Por lo tanto la distribución posterior es la siguiente:

$$f_B(x_1, ..., x_d) \propto exp[-x_1^2/200 - \frac{1}{2}(x_2 + Bx_1^2 - 100B)^2 - \frac{1}{2}(x_3^2 + x_4^2 + ... + x_d^2)] \quad (5)$$

Se probaron dos experimentos, en el primero con dos dimensiones para observar el comportamiento de la cadea en el mapa de contorno, el segundo experimento consiste en probar con 50 variables y 100'000 iteraciones. En la figura 3 se ilustra el comportamiento de las primeras cinco variables del segundo experimento (parte usperior izquierda) y la primera variable (parte superior derecha) del primer experimento junto con su mapa de contorno (parte inferior izquierda)

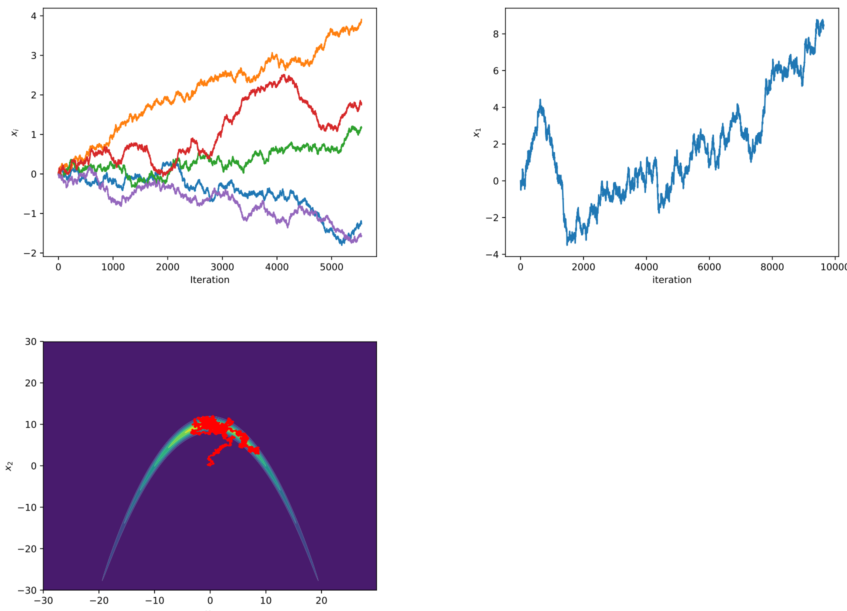

Figura 3: En la parte superior izquierda se muestran las primero cinco variables de ejecutar AM con 50 variables y 100'000 iteraciones. En la parte superior derecha se muestra la primer variable al ejecutar AM con 2 variable y 1'000 iteraciones, su correpondiente mapa de contorno ilustra en la parte inferior izquierda.

.

3. Adaptive Metropolis-Within-Gibbs

En algunos problemas no es posible considerar métodos clásicos, en este apartado se revisa un problema en el que no es posible utilizar Gibbs-sampling en su forma clásica y por lo tanto se hibridiza con una estrategia adaptativa. Considere el siguiente modelo:

$$
\begin{tikzcd}
 & & \mu \arrow[dll] \arrow[d] \arrow[drr] & & \\
\theta_1 \arrow[d] & \cdots & {} \arrow[d] & \cdots & \theta_K \arrow[d] \\
Y_{11},\dots,Y_{1r_1} & & {} & & Y_{K1},\dots,Y_{Kr_K}
\end{tikzcd}
$$

donde

$$
\begin{aligned}
&\theta_i \sim Cauchy(\mu, A) \quad [1 \leq i \leq K] \\
&Y_{ij} \sim N(\theta_i, V) \quad [1 \leq j \leq r_i] \\
&\quad priors: \\
&\mu \sim N(0,1), \quad A, V \sim IG(1,1)
\end{aligned}
\tag{6}
$$

además $IG(a,b)$ es la distribución gamma inversa con densidad proporcional a $e^{-b}x^{-(a+1)}$, y la distribución Cauchy es proporcional a $[1+((x-m)/s)^2]^{-1}$. Este modelo da origen a la distribución posterior $\pi(.)$ en el vector dimensional $(A, V, \mu, \theta_1, ...\theta_k)$, condicionado a los datos observados $_{ij}$. En el artículo se toma $K = 500$ y r_i varía entre 5 y 500 con 200'000 iteraciones de batch y el tamaño del batch de 50 (200'000*50 = 10'000'000 iteraciones). Por cuestiones de tiempo, en este trabajo en su lugar se consideran dimensiones menores y un número de iteraciones menor. El modelo que resulta es muy complicado y debido a que existe la distribución de Cauchy se descompone su conjugado, por lo tanto utilizar un método clásico de Gibbs es infactible. En su lugar se utiliza un algoritmo en el cual se considera actualizar las variables agregando un incremento de $N(0, \sigma^2)$, en esta estrategia se considera la razón de metrópolis de aceptación usual. Primero se define la distribución posterior de la siguiente forma:

$$
\begin{aligned}
f(A,V,\mu,\theta_1,..,\theta_K) &\propto \left[\prod_{i=1}^{K}\prod_{j=1}^{r_i} N(Y_{ij},\theta_i V)\times Cauchy(\mu,A;\theta_i)\right]\\
&\times N(\mu_0,\sigma_0)\times IG(A;a_1,b_1)\times IG(V;a_2,b_2)\\
log(f(A,V,\mu,\theta_1,..,\theta_K)) &\propto \left[\sum_{i=1}^{K}\left(Cauchy(\mu,A;\theta_i)\sum_{j=1}^{r_i} N(Y_{ij},\theta_i V)\right)\right]\\
&\times N(\mu_0,\sigma_0)\times IG(A;a_1,b_1)\times IG(V;a_2,b_2)\\
&\propto \left[\sum_{i=1}^{K}\left(-log(1+((\theta_i-\mu)/A)^2)+\sum_{j=1}^{r_i}[-log(V^{0,5})-\frac{1}{2V}(Y_{ij}-\theta_i)^2]\right)\right]\\
&- log(\sigma_0)-\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2-\frac{b_1}{A}-(a_1+1)log(A)-\frac{b_2}{V}-(a_2+1)log(V)
\end{aligned}
\tag{7}
$$

3.0.1. Algoritmo de Adaptive Metropolis-Within-Gibbs

El algoritmo para la actualización de las varianzas se enlistan a continuación:

1: For each variable i $[i \leq i \leq K+3]$, create a variable ls_i giving the logarithm of the standard deviation.
2: Begin with unit variance $ls_i = 0\ \forall i \in K$.
3: After n^{th} batch of 50 iterations update each ls_i

$$
ls_i = \begin{cases} +\delta(n), & \text{if } Acceptance \quad rate \quad batch > 0{,}44 \\ -\delta(n), & \text{otherwise} \end{cases} \tag{8}
$$

4: Choose a $\delta(n) \to 0$ to satisfy Diminishing Adaptation condition (e.g. $\delta(n) = min(0{,}001, n^{-0,5})$).
5: Restrict each $ls_i \in [-M, M]$ to satisfy Boundary Convergence condition.

3.1. Validación Experimental Metropolis-Within-Gibbs

Se tomaron valores r_i aleatoreamente del conjunto $\{5,3,2,1\}$, el tamaño de bathc considerado es de 50 y el número de batch totales es de 5000. En la figura 4 se muestran los valores obtenido del logaritmo de std para cada una de las primero tres variables θ_i, se puede observar que existe una convergencia en las primeras 100 iteraciones. En la figura 5 se muestra que el valor de las variables que igualmente convergen aproximadamente en las primeras 100 iteraciones. En la tabla 1 se muestran los valores obtenidos considerando la distancia promedio de los desplazamientos

Variable	**Adaptive**	**Fixed**
θ_1	4.27	9.98
θ_2	4.08	9.73
θ_3	4.69	10.74
θ_4	4.27	10.75
θ_5	4.31	9.63

Cuadro 1: Valores obtenidos con la distancia promedio de los desplazamientos

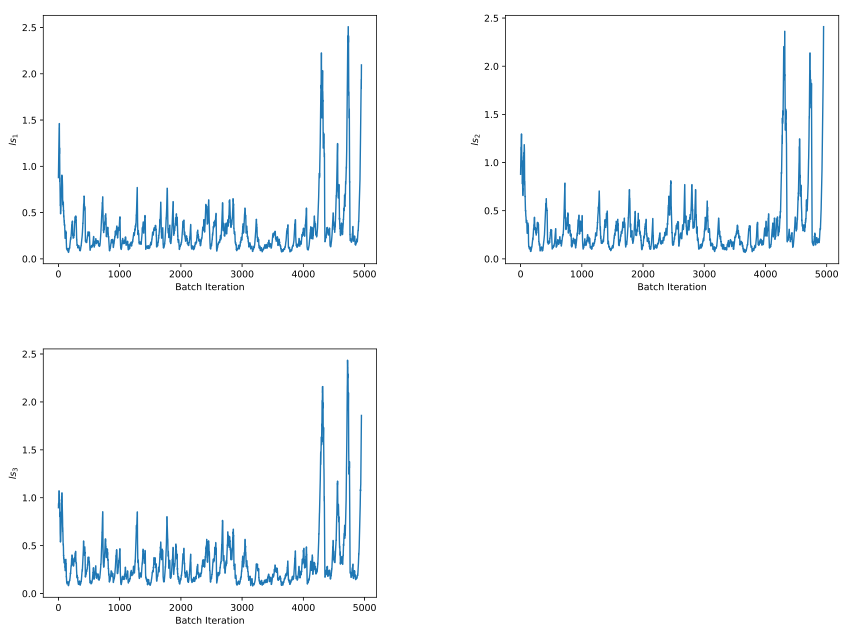

Figura 4: Logaritmo de std de las primero tres variables θ_i.

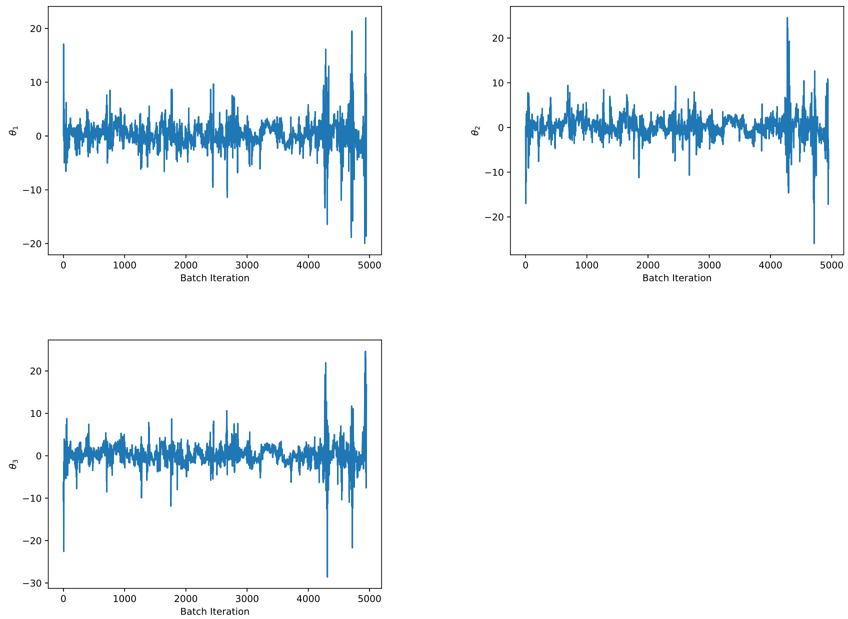

Figura 5: Valores de las primero tres variables θ_i.

4. Conclusions

En los últimos años los algoritmos MCMC ha surgido un mayor interés en el diseño y desarrlo de distintos esquemas, esto principalmente por el incremento de poder computacional. Realizar la simulación de distribuciones complejas es un problema que surje principalmente en problemas aplicados. En este trabajo se implementaron varias estrategias del trabajo de Roberts et. al *Examples Adaptive MCMC*, todo el código se implementó en python. Algunas de las conclusiones más importantes de este proyecto son:

- Los métodos adaptativos MCMC proporcionan buenos resultados para encontrar buenos valores de los hiperparámetro de las distribuciones, como es el caso de la varianza, esto es útil particularmente en altas dimensionalidades cuando no es posible asignar los parámetros a mano.

- Se considera que las estrategias adaptativas usualmente son muy *greedy*, ya que tratan de adaptar la información de forma iterativa (desde la salida anterior), estos algoritmos toman un tiempo considerable para recuperarse y se ven comprometidos al tener información inicial errónea o no confiable.

- Se deberían realizar más trabajo en el camino de temas adaptativos a la par que con la parte teórica.
- Es muy importante evitar el diseño de métodos adaptativos cuyo mecanismo interno sea muy oscuro, y debe existir darse un peso importante de la correspondencia entre la implementación (codificación) y lo explicado en los textos.

Referencias

[1] G. O. Roberts, J. S. Rosenthal, Examples of adaptive mcmc, Journal of Computational and Graphical Statistics 18 (2009) 349–367.

[2] H. Haario, E. Saksman, J. Tamminen, Adaptive proposal distribution for random walk metropolis algorithm, Computational Statistics 14 (1999) 375–396.