



COLEGIO DE POSTGRADUADOS

INSTITUCIÓN DE ENSEÑANZA E INVESTIGACIÓN EN CIENCIAS AGRÍCOLAS

CAMPUS MONTECILLO

POSTGRADO DE SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA
ESTADÍSTICA

MODELOS LINEALES GENERALIZADOS CON RESTRICCIONES LINEALES EN LOS PARÁMETROS DE REGRESIÓN

LUIS COLORADO MARTÍNEZ

T E S I S

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

MONTECILLO, TEXCOCO, EDO. DE MÉXICO
2008

La presente tesis, titulada: **Modelos lineales generalizados con restricciones lineales en los parámetros de regresión**, realizada por el alumno: **Luis Colorado Martínez**, bajo la dirección del Consejo Particular indicado, ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

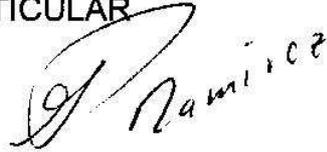
MAESTRO EN CIENCIAS

SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA ESTADÍSTICA

CONSEJO PARTICULAR

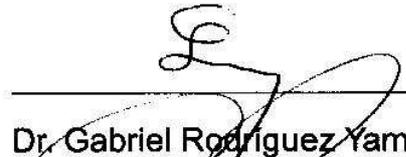
CONSEJO PARTICULAR

CONSEJERO



Dr. Gustavo Ramírez Valverde

DIRECTOR DE TESIS



Dr. Gabriel Rodríguez Yam

ASESOR



Dr. Sergio Pérez Elizalde

Montecillo, Texcoco, Estado de México, Febrero de 2008

MODELOS LINEALES GENERALIZADOS CON RESTRICCIONES LINEALES EN LOS PARÁMETROS DE REGRESIÓN

Luis Colorado Martínez, MC

Colegio de Postgraduados, 2008

Para analizar Modelos Lineales Generalizados (MLG) con parámetro de escala conocido, en el que el vector de coeficientes de regresión se encuentra sujeto a restricciones lineales de desigualdad, en este trabajo se propone un método Bayesiano, denominado Gibbs Transformado con Muestreo de Importancia (GTMI). Primero se implementa una cadena de Markov de Monte Carlo (MCMC) a una aproximación normal (truncada) de la distribución final del vector de coeficientes de regresión y luego se utiliza el método de importancia de muestreo, sugerido por Fosdick (1963) y Hastings (1970) para realizar inferencias en los parámetros de regresión. Para muestrear la aproximación normal truncada se propone utilizar el muestreo Gibbs eficiente propuesto por *Rodríguez-Yam (2003)*. Se proporcionan condiciones de regularidad bajo las cuales el método puede implementarse.

Para ilustrar esta metodología se realiza un estudio de simulación comparativo entre los resultados obtenidos con el GTMI y los obtenidos con el método propuesto por Dellaportas y Smith (1993) para un MLG Bernoulli. También se realiza un estudio en el que sólo se implementa el GTMI para un MLG exponencial.

Basándose en los resultados del estudio de simulación para el MLG Bernoulli, se nota que el método propuesto por Dellaportas y Smith puede converger más lento que el GTMI. Además, con el GTMI se obtienen muestras “casi” independientes y las estimaciones para la media de la distribución final del vector de coeficientes de regresión tienen menor sesgo. En este estudio se corrobora el hecho de que la implementación directa del muestreo Gibbs en un problema de restricción puede resultar ineficiente. En los resultados del estudio de simulación para el MLG exponencial, se nota que el GTMI es de convergencia rápida y se obtiene sesgo pequeño para el estimador de la media en todos los casos.

Palabras clave: Modelos lineales generalizados, restricciones lineales de desigualdad, distribución normal multivariada truncada, muestreo Gibbs, aproximación normal, muestreo de importancia.

GENERALIZED LINEAR MODELS WITH LINEAR CONSTRAINTS IN THE REGRESSION PARAMETERS

Luis Colorado Martínez, MC
Colegio de Postgraduados, 2008

We propose a Bayesian method called Transformed Gibbs with Importance Sampling (TGIS) to analyze Generalized Linear Models (GLM) with a known linear parameter, where the vector of regression coefficients has inequality linear constraints. Firstly, a Markov Chain Monte Carlo (MCMC) is implemented onto a normal approximation (truncated) of the final distribution of the vector of regression coefficients, and then we use the importance sampling method suggested by Fosdick (1963) and Hastings (1970) to make inferences from the regression coefficients. We propose the use of the efficient Gibbs sampler proposed by Rodriguez-Yam (2003) to sample the truncated normal approximation. Regularity conditions where the method can be implemented are given. To illustrate this methodology, a comparative simulation study is done with the results obtained by the TGIS and those obtained from the method proposed by Dellaportas and Smith (1993) for a Bernoulli GLM. Another study is done solely implementing TGIS for an exponential GLM. Based on the results of the simulation study for the Bernoulli GLM, we see that the method proposed by Dellaportas and Smith can converge more slowly than the TGIS. Moreover, with the TGIS we draw almost independent samples, and the estimates for the mean final distribution of the vector of regression coefficients are less biased. This study corroborates the fact that the direct implementation of the Gibbs sampler in a constraint problem may turn out to be inefficient. The results of the simulation study for the exponential GLM show that TGIS has a rapid convergence and a small bias for the mean estimator in all cases.

Key words: Generalized linear models, Inequality linear constraints, Truncated multivariate normal distribution, Gibbs sampler, Importance sampling.

A mi esposa *Carmina*, a mi hija *Mónica* y a mi hijo *Luis Alberto*... y

A mis hermanos: *Julio Alberto*, *Armando* y *Gustavo*.

AGRADECIMIENTOS

A la *Universidad Autónoma de Yucatán (UADY)* y al *Programa de Mejoramiento del Profesorado (PROMEP)* por el apoyo económico otorgado para la realización de mis estudios de maestría.

A los integrantes de mi Consejo Particular:

Dr. Gabriel Rodríguez Yam, por su excelente dirección para la realización de este trabajo.

Dr. Gustavo Ramírez Valverde, por sus observaciones, consejos y apoyo otorgados para la realización de este trabajo.

Dr. Sergio Pérez Elizalde, por la disposición y aportes realizados en el desarrollo de este trabajo.

Al *Dr. Humberto Vaquera Huerta*, por el apoyo logístico otorgado para la obtención de los resultados en el estudio de simulación.

Al *M.C. Paulino Pérez Rodríguez*, por su valiosa ayuda en el uso apropiado de los softwares R y Latex.

A mis profesores y todos aquellos que de alguna u otra manera fueron coparticipes de esta tarea.

CONTENIDO

1. INTRODUCCIÓN	1
2. OBJETIVOS	3
2.1. Objetivo general	3
2.2. Objetivos específicos	3
3. REVISIÓN DE LITERATURA	4
3.1. Modelos lineales generalizados	4
3.2. El modelo bayesiano	6
3.3. Cadena de Markov Monte Carlo	7
3.3.1. Monitoreo de convergencia	8
3.3.1.1. Diagnóstico de convergencia de Gelman y Rubin	8
3.3.1.2. Diagnóstico de convergencia de Reftery y Lewis	10
3.4. Algoritmo de muestreo Gibbs	10
3.5. Algoritmo de muestreo Gibbs-Dellaportas y Smith (GDS)	11
3.5.1. Algoritmo ARS (Muestreo de Rechazo Adaptativo)	11
3.5.1.1. Metodología	11
3.5.1.2. Implementación	13
3.5.2. Metodología	13
3.5.3. Implementación	14
3.6. Muestreo Gibbs-Rodríguez Yam (MGRY)	14
3.6.1. Metodología	15
3.6.2. Implementación	16
3.7. Muestreo de Importancia (IS)	16
3.8. Aproximación normal de la verosimilitud	17
3.9. Otros resultados	18
4. MATERIALES Y MÉTODOS	20
4.1. Planteamiento del modelo bayesiano	21
4.2. Gibbs transformado con muestreo de importancia (GTMI)	22
4.2.1. Aproximación normal de $L(\boldsymbol{\beta}; \mathbf{y})$	22
4.2.2. Planteamiento aproximado para el modelo Bayesiano $\pi(\boldsymbol{\beta} \mathbf{y})$	24
4.2.2.1. Distribución inicial-Uniforme	24

4.2.2.2. Distribución inicial-Normal multivariada	25
4.2.3. Muestreo de la distribución $\pi_a(\boldsymbol{\beta} \mathbf{y})$	25
4.2.4. Inferencias para funciones del vector $\boldsymbol{\beta}$	25
4.2.5. Condiciones de regularidad	27
4.2.5.1. Para la existencia y obtención de la moda	27
4.2.5.2. Para la aproximación normal	29
5. RESULTADOS	30
5.1. Distribución Bernoulli	30
5.1.1. Caso 1: $\boldsymbol{\beta} = (4, 3)^t$	31
5.1.1.1. Resultados para $n = 30$	31
5.1.1.2. Resultados para $n = 100$	37
5.1.2. Caso 2: $\boldsymbol{\beta} = (4.5, 8.3)^t$	39
5.1.2.1. Resultados para $n = 30$ y $n = 100$	39
5.1.3. Caso 3: $\boldsymbol{\beta} = (6.2, 0.8)^t$	41
5.1.3.1. Resultados para $n = 30$ y $n = 100$	41
5.2. Distribución Exponencial	44
5.2.1. Caso 1: $\boldsymbol{\beta} = (3.09, 4.05, 4.30, 2.52, 0.85)^t$	44
5.2.1.1. Resultados para $n = 30$	44
5.2.1.2. Resultados para $n = 100$ y $n = 400$	49
5.2.2. Caso 2: $\boldsymbol{\beta} = (2.32, 2.74, 1.01, 2.57, 0.79)^t$	49
5.2.2.1. Resultados para $n = 30$, $n = 100$ y $n = 400$	49
6. CONCLUSIONES	51
7. DISCUSIÓN	53
8. REFERENCIAS	54
ANEXO A. PROGRAMAS EN R	58
A1. Simulación de datos	58
A2. Elección de puntos de inicio	59
A3. Implementación del GTMI	61
A4. Implementación del GDS	64
A5. Gráficas	69
A6. Estadísticas	76

LISTA DE CUADROS

1.	Covariables y realizaciones, Bernoulli caso 1, $n = 30$	32
2.	Diagnóstico de Raftery y Lewis para β_1 y β_2 , Bernoulli caso 1, $n = 30$	34
3.	Estimación, desviación estándar y sesgo, Bernoulli caso 1, $n = 30$	36
4.	Estimación, desviación estándar y sesgo, Bernoulli caso 1, $n = 100$	38
5.	Vector de medias, matriz de covarianzas y matriz de correlaciones para $\pi_a(\boldsymbol{\beta} \mathbf{y})$, Bernoulli caso 2, $n = 30$ y $n = 100$	39
6.	Estimación, desviación estándar y sesgo, Bernoulli caso 2, $n = 30$ y $n = 100$	41
7.	Vector de medias, matriz de covarianzas y matriz de correlaciones para $\pi_a(\boldsymbol{\beta} \mathbf{y})$, Bernoulli caso 3, $n = 30$ y $n = 100$	41
8.	Estimación, desviación estándar y sesgo, Bernoulli caso 3, $n = 30$ y $n = 100$	43
9.	Covariables y realizaciones, exponencial caso 1, $n = 30$	45
10.	Diagnóstico de Raftery y Lewis para las componentes de $\boldsymbol{\beta}$, exponencial caso 1, $n = 30$	47
11.	Estimación, desviación estándar y sesgo, exponencial caso 1, $n = 30$	48
12.	Vector de medias y matriz de covarianzas para $\pi_a(\boldsymbol{\beta} \mathbf{y})$, exponencial caso 1, $n = 100$ y $n = 400$	49
13.	Estimación, desviación estándar y sesgo, exponencial caso 1, $n = 100$ y $n = 400$	49
14.	Vector de medias y matriz de covarianzas para $\pi_a(\boldsymbol{\beta} \mathbf{y})$, exponencial caso 2, $n = 30$, $n = 100$ y $n = 400$	50
15.	Media, desviación estándar y sesgo, exponencial caso 2, $n = 30$, $n = 100$ y $n = 400$	50

LISTA DE FIGURAS

1.	Región de truncamiento, caso Bernoulli	31
2.	Contorno de la distribución final y aproximada, Bernoulli caso 1, $n = 30$	33
3.	Primeras 500 iteraciones de cada cadena para los métodos GTMI y GDS, Bernoulli caso 1, $n = 30$	34
4.	Factor de escala para β_1 y β_2 por método, Bernoulli caso 1, $n = 30$	35
5.	Autocorrelaciones para β_1 y β_2 obtenida con el GTMI, Bernoulli caso 1, $n = 30$. . .	36
6.	Autocorrelaciones para β_1 y β_2 obtenida con el GDS, Bernoulli caso 1, $n = 30$. . .	36
7.	Factor de escala para β_1 y β_2 por método, Bernoulli caso 1, $n = 100$	37
8.	Autocorrelaciones para β_1 y β_2 obtenida con el GTMI, Bernoulli caso 1, $n = 100$. .	38
9.	Autocorrelaciones para β_1 y β_2 obtenida con el GDS, Bernoulli caso 1, $n = 100$. . .	38
10.	Primeras 500 iteraciones de cada cadena para los métodos GTMI y GDS, Bernoulli caso 2, $n = 30$	39
11.	Autocorrelaciones para β_1 y β_2 obtenida con el GTMI, Bernoulli caso 2, $n = 30$. . .	40
12.	Autocorrelaciones para β_1 y β_2 obtenida con el GDS, Bernoulli caso 2, $n = 30$. . .	40
13.	Primeras 500 iteraciones de cada cadena para los métodos GTMI y GDS, Bernoulli caso 3, $n = 30$	42
14.	Autocorrelaciones para β_1 y β_2 obtenida con el GTMI, Bernoulli caso 3, $n = 30$. . .	42
15.	Autocorrelaciones para β_1 y β_2 obtenida con el GDS, Bernoulli caso 3, $n = 30$. . .	43
16.	Factor de escala para las componentes de β , exponencial caso 1, $n = 30$	47
17.	Autocorrelaciones para las componentes de β obtenida con el GTMI, exponencial caso 1, $n = 30$	48

1. INTRODUCCIÓN

El problema de parámetros restringidos surge en diversas aplicaciones como en datos categóricos ordinales, superficies de respuesta, pruebas de desarrollo en confiabilidad, componentes de varianza, bioensayos, graduación actuarial, etc (Gelfand *et al.*, 1992). En un sentido más amplio las restricciones podrían presentarse junto con datos truncados y dependencia entre los parámetros, haciendo el análisis de los datos sumamente difícil. La implementación directa de los algoritmos de muestreo Gibbs (Geman y Geman, 1984; Gelfand y Smith, 1990) y Metropolis-Hastings (Metropolis *et al.*, 1953; Hastings, 1970) para muestrear el modelo resultante, es casi imposible o en su defecto la implementación resulta ser computacionalmente muy ineficiente. Desde el punto de vista clásico el problema de la restricción en los parámetros se ha abordado desde varias perspectivas (Liew y Shim, 1976; Chaudhuri y Perlman, 2003); sin embargo, las implementaciones tienen serias limitaciones cuando las restricciones son arbitrarias, el número de parámetros es muy grande o cuando la probabilidad de la región de truncamiento con respecto a la probabilidad de todo el espacio paramétrico es muy pequeña. Un trabajo con restricciones de orden en los parámetros en modelos de regresión lineal es el de Mukerjee y Tu (1995).

Para el caso de los MLG con parámetros de regresión restringidos, desde el punto de vista clásico, McDonald y Diamond (1990), consideran el problema de encontrar los estimadores de máxima verosimilitud cuando algunos o todos los parámetros de regresión están sujetos a ser mayores o iguales a cero. Para su análisis, derivan las condiciones de Kuhn-Tucker y las utilizan para caracterizar la solución de este problema de estimación restringida, cuando el estimador de máxima verosimilitud existe y es único. Geyer (1991) usa el bootstrap paramétrico para obtener el estimador de máxima verosimilitud cuando los parámetros de regresión están sujetos a restricciones de orden.

En Gelfand *et al.*, (1992) se analiza el problema de los parámetros restringidos desde el punto de vista Bayesiano, bajo el supuesto de que las funciones de distribución correspondientes a las condicionales completas son conocidas. En Chen *et al.*, (2000) se presenta una metodología que permite obtener inferencias Bayesianas para un modelo con parámetros restringidos, la cual está basada en la idea de *mezclas reponderadas* de Geyer (1994).

Por otra parte, Rodríguez-Yam (2003), propuso una implementación eficiente del muestreo Gibbs en un modelo Bayesiano de regresión lineal con parámetros de regresión sujetos a un conjunto de restricciones lineales de desigualdad e igualdad, y a diferencia del muestreo Gibbs propuesto por Geweke (1996), el conjunto de restricciones puede ser linealmente dependiente y el número de restricciones lineales puede exceder la dimensión del vector de coeficientes de regresión.

Desde un enfoque Bayesiano, Dunson y Neelon (2002) consideran el caso cuando los parámetros de regresión están sujetos a restricciones de orden. Argumentan que la selección de una distribución inicial con soporte en el espacio restringido complica el análisis, por lo que proponen el mapeo de una densidad final no restringida usando una transformación con regresión isotónica.

Existen otros trabajos con enfoque Bayesiano para los MLG, pero sin contemplar el problema de la restricción en los parámetros de regresión; por ejemplo Zeger y Rezaul (1991) y Dellaportas y Smith (1993).

En este trabajo se propone el Método Gibbs Transformado con Muestreo de Importancia (GTMI) para tratar con el problema de la restricción lineal con desigualdades en el vector de coeficientes de regresión, en un MLG con parámetro de escala conocido, y desde un punto de vista Bayesiano. Para su implementación se utiliza el algoritmo de muestreo Gibbs de Rodríguez-Yam (2003) y el algoritmo IS (por sus siglas en inglés, Importance Sampling). Se realiza un estudio de simulación para un MLG Bernoulli, en el que se comparan los resultados obtenidos por el GTMI y el propuesto por Dellaportas y Smith (1993). También se realiza un estudio de simulación para un MLG exponencial en el que solo se implementa el GTMI.

2. OBJETIVOS

2.1. Objetivo general

Plantear un modelo Bayesiano e implementar un método Gibbs de muestreo para un MLG con parámetro de escala conocido, en el que los coeficientes de regresión se encuentran sujetos a un conjunto de restricciones lineales de desigualdad.

2.2. Objetivos específicos

- a) Plantear un modelo Bayesiano para el vector de coeficientes de regresión.
- b) Implementar un algoritmo de muestreo Gibbs para realizar inferencias sobre el vector de coeficientes de regresión.
- c) Proporcionar condiciones de regularidad, bajo las cuales se obtengan buenos resultados con el algoritmo de muestreo Gibbs en (b).
- d) Comparar los resultados que se obtienen con el método en (b) y con los del método propuesto por Dellaportas y Smith (1993), en un estudio de simulación para el MLG Bernoulli con tamaños de muestra 30 y 100.
- e) Realizar un estudio de simulación en un MLG exponencial, donde se implemente solo el método en (b) con tamaños de muestra 30, 100 y 400.

3. REVISIÓN DE LITERATURA

En esta Sección se introducen conceptos de la teoría de los MLG, modelación Bayesiana y cadenas de Markov; así como metodologías y resultados obtenidos en la literatura que son útiles para alcanzar los objetivos de este trabajo.

3.1. Modelos lineales generalizados

Considérese la variable aleatoria y cuya función de densidad depende de un sólo parámetro θ . Se dice que la distribución pertenece a la familia exponencial si ésta puede escribirse de la forma:

$$f(y; \theta) = \exp[a(y)b(\theta) + c(\theta) + d(y)], \quad (1)$$

donde $a(\cdot)$, $b(\cdot)$, $c(\cdot)$ y $d(\cdot)$ son funciones reales conocidas. Si $a(y) = y$, se dirá que la familia tiene forma canónica (estándar) y $b(\theta)$ es llamado el parámetro natural de la distribución. Si hay otros parámetros que forman parte de las funciones $a(\cdot)$, $b(\cdot)$, $c(\cdot)$ y $d(\cdot)$, en adición al parámetro θ , éstos son tratados como si fueran conocidos (Dobson, 1945). Además,

$$\begin{aligned} E[a(y)] &= \frac{-c'(\theta)}{b'(\theta)} \\ \text{var}[a(y)] &= \frac{b''(\theta)c'(\theta) - c''(\theta)b'(\theta)}{[b'(\theta)]^3}. \end{aligned} \quad (2)$$

La forma canónica de varios modelos importantes se pueden escribir como (1), por ejemplo en el modelo Poisson se tiene que $b(\theta) = \log\theta$, $c(\theta) = -\theta$ y $d(y) = -\log y!$. En el modelo Binomial, $b(\pi) = \log(\pi/1 - \pi)$, $c(\pi) = n\log(1 - \pi)$ y $d(y) = \log\binom{n}{y}$ y en el modelo normal, $b(\mu, \sigma) = \mu/\sigma^2$, $c(\mu, \sigma) = -\mu^2/2\sigma^2 - (1/2)\log(2\pi\sigma^2)$ y $d(y) = -y^2/2\sigma^2$. Otros modelos importantes correspondientes a las distribuciones Geométrica, Binomial negativa, Exponencial, Gamma y la Inversa Gaussiana son casos especiales de (1).

La unificación de muchos métodos estadísticos usando la idea de un MLG fue demostrado por Nelder y Wedderburn (1972). Este modelo se define en términos de un conjunto de variables independientes y_1, \dots, y_n , en el que cada una de estas variables aleatorias tiene una función de densidad con las siguientes propiedades:

1. La función de densidad de cada y_i tiene la forma canónica y depende de un solo parámetro θ_i (las θ_i 's no tienen por que ser las mismas), así

$$f(y_i; \theta_i) = \exp[y_i b(\theta_i) + c(\theta_i) + d(y_i)] \quad (3)$$

2. La función de densidad de todas las y_i 's tienen la misma forma (por ejemplo todas son normales o binomiales) por lo que los subíndices en b , c y d no son necesarios.

De 1 y 2 se sigue entonces que la función de densidad conjunta de y_1, \dots, y_n es

$$f(y_1, \dots, y_n; \theta_1, \dots, \theta_n) = \exp \left[\sum_{i=1}^n y_i b(\theta_i) + \sum_{i=1}^n c(\theta_i) + \sum_{i=1}^n d(y_i) \right].$$

Típicamente los parámetros θ_i no son de un interés directo (ya que puede haber uno para cada observación). Para la especificación del modelo, usualmente se está interesado en un conjunto más pequeño de parámetros β_1, \dots, β_p (donde $p < n$). Supóngase que $E(y_i) = \mu_i$ donde μ_i es alguna función de θ_i . Para un MLG existe una transformación de μ_i tal que

$$g(\mu_i) = \mathbf{x}_i^t \boldsymbol{\beta}, \quad (4)$$

donde $g(\cdot)$ es una función monótona diferenciable llamada la función liga; \mathbf{x}_i es un vector de variables explicativas (covariables y variables dummy para niveles de factores) de orden p , $\mathbf{x}_i^t = (x_{i1}, \dots, x_{ip})$ y $\boldsymbol{\beta}$ es el vector de coeficientes de regresión de orden p , $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^t$.

Note que:

1. Por (2) y (4) se tiene que $g\left(\frac{-c'(\theta_i)}{b'(\theta_i)}\right) = \mathbf{x}_i^t \boldsymbol{\beta}$, por lo que existe una función $h(\cdot)$ estrictamente creciente y suficientemente suave tal que,

$$\theta_i = h(\mathbf{x}_i^t \boldsymbol{\beta}). \quad (5)$$

2. Si se elije $g(\cdot)$ tal que $\theta_i = \mathbf{x}_i^t \boldsymbol{\beta}$, entonces se dice que $g(\cdot)$ es la función liga canónica correspondiente al modelo. Aunque la liga canónica conduce a propiedades estadísticas deseables para el modelo, ésta no debe reemplazar la calidad de ajuste como un criterio para seleccionar un modelo (McCullagh y Nelder, 1989).

En resumen, un MLG tiene tres componentes:

1. Variables respuestas y_i, \dots, y_n , las cuales se asumen independientes y tienen la misma función de densidad en la familia exponencial. Esta componente es conocida como componente aleatoria.
2. Un conjunto de parámetros $\boldsymbol{\beta}$ y variables regresoras

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^t \\ \vdots \\ \mathbf{x}_n^t \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}$$

3. Una función liga monótona $g(\cdot)$, tal que $g(\mu_i) = \mathbf{x}_i^t \boldsymbol{\beta}$, donde $\mu_i = E(y_i)$. Esta componente es conocida como componente sistemática.

3.2. El modelo bayesiano

Un modelo Bayesiano está basado en la especificación de un modelo de probabilidad para los datos observados D , dado un vector de parámetros desconocidos $\boldsymbol{\theta}$, encabezado por la función de verosimilitud $L(\boldsymbol{\theta}|D)$. Se asume que $\boldsymbol{\theta}$ es aleatorio y tiene una distribución inicial denotada por $\pi(\boldsymbol{\theta})$. Cualquier inferencia sobre $\boldsymbol{\theta}$ está basada en su distribución final, la cual, por el Teorema de Bayes está dada por

$$\pi(\boldsymbol{\theta}|D) = \frac{L(\boldsymbol{\theta}|D) \pi(\boldsymbol{\theta})}{\int_{\Omega} L(\boldsymbol{\theta}|D) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}}, \quad (6)$$

donde Ω denota el espacio paramétrico de $\boldsymbol{\theta}$ y $L(\boldsymbol{\theta}|D)$ es la función de verosimilitud. De (6) es claro que $\pi(\boldsymbol{\theta}|D) \propto L(\boldsymbol{\theta}|D) \pi(\boldsymbol{\theta})$ y así, esto involucra una contribución de los datos observados a través de $L(\boldsymbol{\theta}|D)$, y una contribución de información inicial cuantificada a través de $\pi(\boldsymbol{\theta})$. La cantidad $m(D) = \int_{\Omega} L(\boldsymbol{\theta}|D) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}$ es la constante de normalización de $\pi(\boldsymbol{\theta}|D)$, y es llamada frecuentemente la distribución marginal de los datos o la distribución predictiva inicial.

En muchos modelos ocurre que $m(D)$ no tiene una forma analítica cerrada; este dilema conduce a la pregunta: ¿cómo muestrear de la distribución multivariable $\pi(\boldsymbol{\theta}|D)$, cuando ésta no tiene una forma cerrada?; esta pregunta a la vez conduce a una vasta literatura sobre métodos computacionales para muestrear $\pi(\boldsymbol{\theta}|D)$ y métodos para estimar $m(D)$. Uno de los métodos computacionales más populares para muestrear $\pi(\boldsymbol{\theta}|D)$ es el Gibbs sampler, el cual está basado en un algoritmo de simulación muy poderoso que permite muestrear la distribución $\pi(\boldsymbol{\theta}|D)$ sin necesidad de conocer la constante de normalización $m(D)$.

Inferencias con respecto al vector $\boldsymbol{\theta}$ se realizan a través de la estimación de $E[h(\boldsymbol{\theta})]$, donde $h(\boldsymbol{\theta})$ es una función del vector $\boldsymbol{\theta}$ y la esperanza se calcula con respecto a la distribución final de $\boldsymbol{\beta}$ en (6). Con $E[h(\boldsymbol{\theta})]$ se obtienen inferencias para:

- a) la media de $\boldsymbol{\theta}$, si $h(\boldsymbol{\theta}) = \boldsymbol{\theta}$.
- b) la covarianza de θ_i y θ_j , si $h(\boldsymbol{\theta}) = (\theta_i - E(\theta_i|D))(\theta_j - E(\theta_j|D))$, donde $E(\theta_j|D) = \int_{\Omega} \theta_j \pi(\boldsymbol{\theta}|D) d\boldsymbol{\theta}$, y la integral es con respecto al resto de las otras componentes θ_j .
- c) la densidad predictiva, si $h(\boldsymbol{\theta}) = f(z|\boldsymbol{\theta})$, donde $f(z|\boldsymbol{\theta})$ es la densidad predictiva dado el vector de parámetros $\boldsymbol{\theta}$.
- d) la probabilidad de un conjunto A , si $h(\boldsymbol{\theta}) = I_A(\boldsymbol{\theta})$.

Otras cantidades que pueden obtenerse con $E_{\pi_{\boldsymbol{\theta}}}[h(\boldsymbol{\theta})]$ son: las densidades marginales de cada uno o de un conjunto de los componentes del vector $\boldsymbol{\theta}$, las constantes de normalización, los factores de Bayes, densidades finales, cuantiles e intervalos. Para más detalles ver Chen *et al.*, 2000.

3.3. Cadena de Markov Monte Carlo

Una sucesión de vectores aleatorios, $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$, es una cadena de Markov, si para cada $t \geq 0$, $P(\mathbf{x}_{t+1}|\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{x}_t)$ depende solamente del estado actual \mathbf{x}_t de la cadena (Gilks *et al.*, 1996), i.e., $P(\mathbf{x}_{t+1}|\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{x}_t) = P(\mathbf{x}_{t+1}|\mathbf{x}_t)$. A $P(\cdot|\cdot)$ se le conoce como el kernel de transición de la cadena. En esta definición se asume que $P(\cdot|\cdot)$ no depende de t .

Una cadena de Markov converge a una distribución estacionaria $\varphi(\cdot)$, si $P^t(\cdot|\mathbf{x}_0)$ converge eventualmente a $\varphi(\cdot)$. Es decir, la cadena olvida gradualmente su estado inicial y no depende de t .

Para que la distribución de \mathbf{x}_t converja a una distribución estacionaria, la cadena debe tener tres propiedades importantes (Gilks *et al.*, 1996).

La primera es que la cadena sea *irreducible*, es decir, que para toda i, j exista $t > 0$ tal que $P^t(\mathbf{x}_t = j|\mathbf{x}_0 = i) > 0$.

La segunda es que la cadena debe ser *no periódica*, es decir, si para cada i se cumple que el máximo común divisor de $\{t > 0 : P^t(\mathbf{x}_t = i|\mathbf{x}_0 = i) > 0\}$ es igual a 1.

La tercera y más importante es que la cadena debe ser *recurrente positiva*. La cadena es *recurrente* si para toda i, j se cumple que:

$$\sum_t P^t(\mathbf{x}_t = j|\mathbf{x}_0 = i) = \infty$$

y es *positiva* si existe una distribución $\varphi(\cdot)$, tal que:

$$\sum_i \varphi(i) P^t(\mathbf{x}_t = j | \mathbf{x}_0 = i) = \varphi(j) \quad \forall j, t \geq 0$$

Estas tres propiedades son referidas como condiciones de regularidad de una cadena de Markov.

Para que un algoritmo que utiliza cadenas de Markov de Monte Carlo sea exitoso, debe cumplir dos condiciones. La primera es construir cadenas a lo largo del tiempo, de tal manera que su distribución estacionaria $\varphi(\cdot)$ sea precisamente la distribución de interés $\pi(\cdot)$, y la segunda es lograr una razón de convergencia rápida hacia la distribución estacionaria.

3.3.1. Monitoreo de convergencia

Una manera de monitorear la convergencia de una cadena de Markov, consiste en generar varias cadenas con la misma longitud pero con diferentes puntos de inicio, y checar en qué momento las cadenas han “olvidado” sus puntos de inicio. Hay muchas formas posibles de comparar cadenas paralelas, en esta Sección se describirán dos métodos de diagnóstico de convergencia, el de Gelman y Rubin (1992) el de Raftery y Lewis (1992).

3.3.1.1. Diagnóstico de convergencia de Gelman y Rubin

En 1992, Gelman y Rubin propusieron una aproximación cuantitativa para monitorear la convergencia de una MCMC; el método se basa en monitorear por separado la convergencia de todas las cantidades de interés de una distribución y cuando la varianza entre las diferentes cadenas no es más grande que la varianza dentro de cada cadena individual, se declara convergencia.

Sea $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)$ un vector de parámetros con cierta distribución final, supóngase que se está interesado en cada uno de los componentes θ_i , $i = 1, \dots, p$. El monitoreo de la convergencia se realiza por separado para cada θ_i ; sin pérdida de generalidad sea θ cualquier θ_i y supóngase que se tienen m cadenas paralelas, cada una con longitud n . Para cada θ se etiquetan las m cadenas paralelas de longitud n con (θ_{ij}) , $j = 1, \dots, n$; $i = 1, \dots, m$ y se obtienen la varianza B , entre las cadenas y la varianza W , dentro de las cadenas como sigue:

$$B = \frac{n}{m-1} \sum_{i=1}^m (\bar{\theta}_i - \bar{\theta}_{..})^2,$$

donde $\bar{\theta}_i = \frac{1}{n} \sum_{j=1}^n \theta_{ij}$, $\bar{\theta}_{..} = \frac{1}{m} \sum_{j=1}^m \bar{\theta}_i$ y

$$W = \frac{1}{m(n-1)} \sum_{i=1}^m \sum_{j=1}^n (\theta_{ij} - \bar{\theta}_i)^2. \quad (7)$$

La varianza $\text{var}(\theta|y)$ de la marginal de θ que se estima como el promedio ponderado de W y B , i.e.,

$$\widehat{\text{var}}(\theta|y) = \frac{n-1}{n}W + \frac{1}{n}B, \quad (8)$$

es un estimador insesgado bajo estacionaridad, pero que sobre estima la $\text{var}(\theta|y)$ asumiendo que la distribución de inicio está apropiadamente sobredispersada. Así, si se tiene una sobredispersión, entonces $\widehat{\text{var}}(\theta|y)$ es un estimador *conservador* de la varianza de θ . Mientras que para alguna n finita, la varianza dentro de las cadenas, W subestima la varianza de θ , ya que cada cadena tiene menos variabilidad que la distribución de todas las cadenas. En el límite, cuando $n \rightarrow \infty$, $\widehat{\text{var}}(\theta|y)$ y W aproximan a $\text{var}(\theta|y)$ en direcciones opuestas.

Se puede monitorear la convergencia de la cadena de markov, estimando R , el factor por el cual la escala de la distribución actual de θ puede ser reducido si las simulaciones son continuadas indefinidamente. Esta reducción potencial de escala puede ser estimada usando (8) y (7), de la siguiente manera:

$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}(\theta)}{W}} \quad (9)$$

y es denominado “*estimador de reducción potencial de escala*”.

Cuando la cadena converge, \hat{R} se acerca a 1, significando que las cadenas de Markov están esencialmente sobrepuestas y provienen de la misma distribución. Si \hat{R} es mayor que 1, entonces se tendrá razón para creer que procediendo con más simulaciones se pueden mejorar las inferencias sobre θ .

Se recomienda calcular el *estimador de reducción potencial de escala* para todas las cantidades de interés, descartando la primera mitad de las simulaciones de cada cadena. Si \hat{R} no está cercano a 1 para todos los componentes de interés, una buena idea es continuar la simulación. En la práctica se corren simulaciones hasta que los valores de \hat{R} son todos menores que 1.1 o 1.2.

Una vez que \hat{R} esté cercano a 1 para todas las cantidades de interés, las inferencias se realizan con la muestra obtenida al mezclar las simulaciones de las segundas mitades de las m cadenas paralelas. Las estimaciones son confiables si están basadas en cadenas múltiples con puntos de inicio sobredispersados. Las estimaciones obtenidas antes de la convergencia serán conservadoras y una vez que la convergencia ha sido alcanzada, éstas serán más precisas. Para más detalles ver también Gelman, *et al.*, (2004).

3.3.1.2. Diagnóstico de convergencia de Raftery y Lewis

Este método de convergencia, introducido por Raftery y Lewis (1992), se basa en un criterio de exactitud de la estimación del cuantil q de una función real del vector $\boldsymbol{\theta}$. Se determina el número de iteraciones N que se requiere para estimar el cuantil q dentro de una exactitud de $\pm r$ con probabilidad s . También se obtienen M , N_m e I , donde M es el número de iteraciones iniciales que deben ser descartadas (“burn-in”), N_m es el tamaño de muestra mínimo basado en una muestra aleatoria e I es el factor de dependencia que estima la magnitud en la cual la autocorrelación incrementa el tamaño de muestra requerido, y se calcula como $I = (M + N)/N_m$. Valores de I mayores que 5 indican una fuerte autocorrelación, la cual puede ser resultado de una pobre elección del valor de inicio de la cadena o una alta correlación entre los componentes del vector $\boldsymbol{\theta}$, relativa a su distribución final.

El tamaño de muestra estimado para una variable U está basada en el proceso $Z_t = d(U_t \leq u)$ donde d es la función indicadora y u es el cuantil q de U . El proceso Z_t es derivado de la cadena de Markov de los datos por marginalización y truncamiento, pero no es por sí misma una cadena de Markov. Sin embargo, Z_t puede comportarse como una cadena de Markov si ésta está suficientemente dispersa. Para calcular el valor más pequeño k , el cual hace que Z_t^k se comporte como una cadena de Markov se utiliza el programa “*raftery.diag*”, el cual está basado en el programa FORTRAN “*gibbsit*”, escrito por Lewis (1995) y está disponible en el archivo Statlib.

El criterio para determinar el número de iteraciones “burn-in” m , es que la distribución condicional de Z_m^k dado Z_0 debe estar dentro de la distribución equilibrio de la cadena Z_t^k .

3.4. Algoritmo de muestreo Gibbs

Supóngase que se desea una muestra aleatoria de la distribución final $p(\boldsymbol{\theta}|D)$, donde D denota los datos y $\boldsymbol{\theta}$ un vector de parámetros pero que es difícil o imposible muestrear de $p(\boldsymbol{\theta}|D)$. El muestreo Gibbs, introducido por Geman y Geman (1984) en el contexto de restauración de imágenes, provee un método para generar muestras de $p(\boldsymbol{\theta}|D)$.

Sea $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_q)$, donde los $\boldsymbol{\theta}_i$'s son unidimensionales o multidimensionales y supóngase que se puede muestrear de las densidades condicionales finales $p(\boldsymbol{\theta}_i|D, \boldsymbol{\theta}_j, j \neq i)$. El muestreo Gibbs genera una cadena de Markov como sigue. Se comienza con $\boldsymbol{\theta}_0 = (\boldsymbol{\theta}_1^{(0)}, \dots, \boldsymbol{\theta}_q^{(0)})$ en el soporte de $p(\boldsymbol{\theta}|D)$. En el tiempo t , la realización $\boldsymbol{\theta}_t = (\boldsymbol{\theta}_1^{(t)}, \dots, \boldsymbol{\theta}_q^{(t)})$ se obtiene como sigue:

Se genera $\boldsymbol{\theta}_1^{(t)}$ de $p(\boldsymbol{\theta}_1|D, \boldsymbol{\theta}_2^{(t-1)}, \boldsymbol{\theta}_3^{(t-1)}, \dots, \boldsymbol{\theta}_q^{(t-1)})$.

Se genera $\boldsymbol{\theta}_2^{(t)}$ de $p(\boldsymbol{\theta}_2|D, \boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_3^{(t-1)}, \dots, \boldsymbol{\theta}_q^{(t-1)})$.

⋮

Se genera $\boldsymbol{\theta}_q^{(t)}$ de $p(\boldsymbol{\theta}_q | D, \boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_2^{(t)}, \dots, \boldsymbol{\theta}_{q-1}^{(t)})$.

Bajo condiciones de regularidad, la cadena de Markov $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_t, \dots\}$ tiene distribución estacionaria $p(\boldsymbol{\theta} | D)$ (Gelfand y Smith, 1990). La razón de convergencia depende de las correlaciones entre los componentes del vector $\boldsymbol{\theta}$ (Gilks y Roberts, 1996).

3.5. Algoritmo de muestreo Gibbs-Dellaportas y Smith (GDS)

Dellaportas y Smith (1993) proponen un muestreo Gibbs para obtener una muestra de la distribución final del vector de coeficientes de regresión en un MLG; para su implementación se utiliza el algoritmo ARS (por sus siglas en inglés, Adaptive Rejection Sampling) para muestrear las condicionales completas finales. Para la implementación de este muestreo Gibbs es necesario que la distribución final del vector de coeficientes de regresión sea log-cóncava (i.e., que el logaritmo de la distribución sea dos veces diferenciable y su segunda derivada sea negativa en todo el soporte de la distribución); lo cual se logra cuando se elige la función liga canónica para la componente sistemática del MLG y una distribución inicial log-cóncava para el vector de coeficientes de regresión. Dellaportas y Smith (1993), proporcionan casos especiales de ligas no canónicas para algunos modelos lineales generalizados en los que la verosimilitud es log-cóncava. A continuación se describe la metodología del algoritmo ARS.

3.5.1. Algoritmo ARS (Muestreo de Rechazo Adaptativo)

La técnica de muestreo ARS es un método propuesto por Gilks y Wild (1992) para muestrear eficientemente funciones log-cóncavas univariadas. Las funciones log-cóncavas forman una familia que incluye una amplia variedad de funciones de densidades, ver Gilks y Wild (1992) y Devroye (1986). El algoritmo ARS está basado en el hecho de que cuando una función log-cóncava es evaluada en un conjunto de puntos de su soporte, entonces las tangentes definen una función lineal que la acota superiormente y las cuerdas una función lineal que la acota inferiormente. Estas funciones son utilizadas para desarrollar un muestreo de rechazo.

3.5.1.1. Metodología

Supóngase que se desea muestrear de la función de densidad $f(x)$ con soporte Ψ , y proporcional a $g(x)$ donde $h(x) = \log g(x)$ es cóncava, es decir, es dos veces diferenciable y la segunda derivada es negativa en todo Ψ . Sean x_1, \dots, x_k puntos ordenados del conjunto Ψ , tal que la moda de $h(x)$ se encuentra entre x_1 y x_k , y $S_k := \{x_i\}_{i=1}^k$ y sean las funciones envolventes inferior $l_k(x)$ y superior

$u_k(x)$ definidas para cualquier punto $x \in [x_i, x_{i+1}]$:

$$l_k(x) = h(x_i) + \frac{x_{i+1} - x_i}{h(x_{i+1}) - h(x_i)}(x - x_i),$$

$$u_k(x) = \begin{cases} h(x_i) + h'(x_i)(x - x_i), & x_i \leq x \leq z_i, \\ h(x_{i+1}) + h'(x_{i+1})(x - x_{i+1}), & z_i < x \leq x_{i+1}, \end{cases}$$

donde $h'(\cdot)$ es la función derivada de $h(\cdot)$ y z_i el punto de intersección entre las tangentes en los puntos $(x_i, h(x_i))$ y $(x_{i+1}, h(x_{i+1}))$, está dado por:

$$z_i = x_i + \frac{h(x_i) - h(x_{i+1}) + h'(x_{i+1})(x_{i+1} - x_i)}{h'(x_{i+1}) - h'(x_i)}.$$

Nótese que $x_i < z_i < x_{i+1}$, $i = 1, \dots, k-1$ y que $z_i < z_{i+1}$, $i = 1, \dots, k-2$, donde $z_0 = 0$ y $z_k = \infty$.

Sean $G_k(x)$ y $F_{G_k}(\cdot)$ la función exponencial normalizada de $u_k(x)$ a lo largo del intervalo $[x_1, x_k]$ y la función de distribución acumulada de la densidad exponencial $G_k(\cdot)$ respectivamente,

$$G_k(x) := \frac{\exp(u_k(x))}{c_u}, \quad x_1 \leq x \leq x_k,$$

$$F_{G_k}(z_i) := \frac{1}{c_u} \sum_{j=0}^i \frac{[\exp(u_k(z_{j+1})) - \exp(u_k(z_j))]}{h'(x_{j+1})},$$

donde

$$c_u = \int_{x_1}^{x_k} \exp(u_k(x)) dx = \sum_{i=0}^{k-1} \frac{[\exp(u_k(z_{i+1})) - \exp(u_k(z_i))]}{h'(x_{i+1})}.$$

Así, un valor x muestreado de la densidad $G_k(x)$ es obtenido por

$$x = z_{[i]} + \frac{1}{h'(x_{i+1})} \log \left[1 + \frac{c_u h'(x_{i+1})(v - F_{G_k}(z_{[i]}))}{\exp(u_k(z_{[i]}))} \right], \quad (10)$$

donde v es una realización de la distribución uniforme en $(0, 1)$ y

$$z_{[i]} = \max\{z_i, i = 1 \dots, k \mid F_{G_k}(z_i) \leq v\}. \quad (11)$$

3.5.1.2. Implementación

Para obtener una muestra aleatoria de la densidad f usando el método ARS, realice los siguientes pasos:

1. Elejir puntos x_1, \dots, x_k de Ψ , tal que la moda de $h(x)$ se encuentre entre x_1 y x_k e inicializar el conjunto $S_k = \{x_i\}_{i=1}^k$.
2. Muestrear v de la distribución uniforme en $(0, 1)$.
3. Determinar $z_{[i]}$ en (11).
4. Obtener x en (10).
5. Muestrear u de la distribución uniforme en $(0, 1)$.
6. Realizar la siguiente prueba: si $u \leq \exp(l_k(x) - u_k(x))$ o si $u \leq \exp(h(x) - u_k(x))$ entonces aceptar x , de otro modo rechazar x .
7. Incluir x al conjunto S_k .
8. Repetir los pasos 2 al 7, hasta alcanzar el número deseado de puntos a muestrear.

Para más detalles sobre el método ARS, ver Gilks y Wild (1992) y Dellaportas y Smith (1993).

3.5.2. Metodología

Sea $L(\boldsymbol{\theta}|D)$ la función de verosimilitud de un MLG, donde $\boldsymbol{\theta}$ es el vector de coeficientes de regresión con distribución inicial $\pi(\boldsymbol{\theta})$; supóngase que $L(\boldsymbol{\theta}|D)$ y $\pi(\boldsymbol{\theta})$ son funciones log-cóncavas y que se tiene interés en muestrear la distribución final $\pi(\boldsymbol{\theta}|D) \propto L(\boldsymbol{\theta}|D)\pi(\boldsymbol{\theta})$.

La log-concavidad de $L(\boldsymbol{\theta}|D)$ implica la log-concavidad de la función $L(\theta_j|\boldsymbol{\theta}_{-j}, D)$, la cual es la verosimilitud condicional de la componente θ_j para valores fijos de las otras componentes $\boldsymbol{\theta}_{-j}$. Análogamente la log-cóncavidad de $\pi(\boldsymbol{\theta})$ implica la log-concavidad de la condicional inicial $\pi(\theta_j|\boldsymbol{\theta}_{-j})$. Así, las condicionales univariadas $\pi(\theta_j|\boldsymbol{\theta}_{-j}, D) \propto L(\theta_j|\boldsymbol{\theta}_{-j}, D)\pi(\theta_j|\boldsymbol{\theta}_{-j})$ son log-cóncavas, ya que la suma de dos funciones log-cóncava es log-cóncava (Rockafellar, 1970).

Bajo estas condiciones se utiliza el Método ARS para muestrear las univariadas $\pi(\theta_j|\boldsymbol{\theta}_{-j}, D)$.

3.5.3. Implementación

Supóngase que $\boldsymbol{\theta} := (\theta_1, \dots, \theta_p)$ y sea $\boldsymbol{\theta}_0 = (\theta_1^{(0)}, \dots, \theta_p^{(0)})$ un valor en el soporte de $\pi(\boldsymbol{\theta} | D)$ el cual servirá como punto de inicio para la implementación del muestreo Gibbs. La realización $\boldsymbol{\theta}_t = (\theta_1^{(t)}, \dots, \theta_p^{(t)})$ en el ciclo t se obtiene recursivamente como sigue:

Obtener $\theta_1^{(t)}$ de $\pi(\theta_1 | \theta_2^{(t-1)}, \dots, \theta_p^{(t-1)}, D)$, implementando el algoritmo ARS.

Obtener $\theta_2^{(t)}$ de $\pi(\theta_2 | \theta_1^{(t)}, \theta_3^{(t-1)}, \dots, \theta_p^{(t-1)}, D)$, implementando el algoritmo ARS.

⋮

Obtener $\theta_p^{(t)}$ de $\pi(\theta_p | \theta_1^{(t)}, \theta_2^{(t)}, \dots, \theta_{p-1}^{(t)}, D)$, implementando el algoritmo ARS.

Para más detalles sobre este método, ver Dellaportas y Smith (1993).

3.6. Muestreo Gibbs-Rodríguez Yam (MGRY)

El algoritmo de muestreo Gibbs de Rodríguez-Yam (2003), propuesto para muestrear eficientemente una distribución normal truncada multivariada, se basa en una transformación ortogonal, es de convergencia rápida y requiere sustancialmente pocas iteraciones en la cadena de Markov. Este algoritmo será denotado con las siglas MGRY para su referencia. La implementación del MGRY se basa en el siguiente resultado:

Resultado 3.6.1. Supóngase que $\mathbf{V} \sim N_{\mathbf{T}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, donde $\mathbf{T} \subset \mathfrak{R}^p$ tiene medida de Lebesgue positiva y $\boldsymbol{\Sigma}$ es positiva definida.

a) Sea $\mathbf{Z} := \mathbf{A}\mathbf{V}$, donde \mathbf{A} es una matriz de rango completo de dimensión $r \times p$ con $r \leq p$, entonces $\mathbf{Z} \sim N_{\mathbf{R}}(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^t)$ y $\mathbf{R} := \{\mathbf{A}\mathbf{v} : \mathbf{v} \in \mathbf{T}\}$.

b) Si se partitionan \mathbf{V} , $\boldsymbol{\mu}$ y $\boldsymbol{\Sigma}$ como

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ V_p \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \mu_p \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_1 \\ \boldsymbol{\Sigma}_1^t & \sigma_{pp} \end{bmatrix},$$

entonces $V_p | \mathbf{V}_1 = \mathbf{v}_1 \sim N_{T_p}(\mu_p^*, \sigma_{pp}^*)$, donde

$$\begin{aligned} \mu_p^* &= \mu_p + \boldsymbol{\Sigma}_1^t \boldsymbol{\Sigma}_{11}^{-1} (\mathbf{v}_1 - \boldsymbol{\mu}_1), \\ \sigma_{pp}^* &= \sigma_{pp} - \boldsymbol{\Sigma}_1^t \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_1, \\ T_p &= \{v_p \in \mathfrak{R} : (\mathbf{v}_1, v_p) \in \mathbf{T}\}. \end{aligned} \tag{12}$$

3.6.1. Metodología

Supóngase que $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, tal que $B\boldsymbol{\beta} \leq \mathbf{b}$; sea $\mathbf{T} := \{\boldsymbol{\beta} \in \Re^p \mid B\boldsymbol{\beta} \leq \mathbf{b}\}$. Se desea muestrear (eficientemente) de $N_{\mathbf{T}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Sea \mathbf{A} una matriz de orden $p \times p$ de rango completo, tal que $\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^t = \mathbf{I}$, donde \mathbf{I} es la matriz identidad de orden $p \times p$. Para simplificar la forma funcional de la distribución normal multivariada truncada, sea \mathbf{Z} el vector de orden p tal que

$$\mathbf{Z} := \mathbf{A}\boldsymbol{\beta}. \quad (13)$$

Aplicando a) del **Resultado 3.6.1** se obtiene

$$\mathbf{Z} \sim N_{\mathbf{S}}(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^t),$$

donde

$$\begin{aligned} \mathbf{S} &:= \{\mathbf{A}\boldsymbol{\beta} : \boldsymbol{\beta} \in \mathbf{T}\} = \{\mathbf{z} \in \Re^p : \mathbf{D}\mathbf{z} \leq \mathbf{b}\}, \\ \mathbf{D} &:= \mathbf{B}\mathbf{A}^{-1}. \end{aligned} \quad (14)$$

Sean

$$\begin{aligned} \boldsymbol{\alpha} &:= \mathbf{A}\boldsymbol{\mu}, \\ \mathbf{Z}_{-j} &:= (Z_1, \dots, Z_{j-1}, Z_{j+1}, \dots, Z_p)^t, \\ \mathbf{z}_{-j} &:= (z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_p)^t. \end{aligned}$$

Aplicando b) del **Resultado 3.6.1** a las diferentes particiones de \mathbf{Z} , $\boldsymbol{\alpha}$, y $\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^t = \mathbf{I}$, obtenida para cada j , se obtiene:

$$Z_j \mid \mathbf{Z}_{-j} = \mathbf{z}_{-j} \sim N_{S_j}(\alpha_j, 1) \quad j = 1, \dots, p, \quad (15)$$

donde el conjunto S_j se obtiene de (12) y (14) como

$$S_j = \{z_j \in \Re : \mathbf{z} \in \mathbf{S}\} = \{z_j \in \Re : \mathbf{D}\mathbf{z} \leq \mathbf{b}\}, \quad j = 1, \dots, p.$$

Para simplificar el conjunto S_j , sea \mathbf{D}_{-j} , la matriz que se obtiene al suprimir la j -ésima columna

de la matriz $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_{j-1}, \mathbf{d}_j, \mathbf{d}_{j+1}, \dots, \mathbf{d}_p]$. El conjunto S_j puede ser obtenido como

$$S_j = \{z_j \in \mathfrak{R} : \mathbf{d}_j z_j \leq \mathbf{b} - \mathbf{D}_{-j} \mathbf{z}_{-j}\}, \quad j = 1, \dots, p. \quad (16)$$

Como $\mathbf{T} := \{\boldsymbol{\beta} \in \mathfrak{R}^p \mid \mathbf{B}\boldsymbol{\beta} \leq \mathbf{b}\}$ es un conjunto convexo de \mathfrak{R}^p , entonces el conjunto S_j en (16) es un intervalo de la forma $[l_j, u_j]$, $(-\infty, u_j]$ o $[l_j, \infty)$, donde los valores l_j y u_j se obtienen de las restricciones en (16).

3.6.2. Implementación

Para obtener una muestra de la distribución $N_{\mathbf{T}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ del vector $\boldsymbol{\beta}$, con el algoritmo MGRY, primero se obtiene una muestra del vector transformado \mathbf{Z} obtenido por (13) y por medio de la identidad $\boldsymbol{\beta} = \mathbf{A}^{-1}\mathbf{Z}$ se obtiene la muestra para el vector $\boldsymbol{\beta}$.

El algoritmo MGRY es el siguiente:

1. Elejir $\boldsymbol{\beta}^0 = [\beta_1^0, \beta_2^0, \dots, \beta_p^0]^t \in \mathbf{T}$ como valor inicial y con este valor determinar el punto de inicio $\mathbf{z}^0 = [z_1^0, z_2^0, \dots, z_p^0]^t \in \mathbf{S}$, donde $\mathbf{z}^0 = \mathbf{A}\boldsymbol{\beta}^0$.
2. Para $j = 1, \dots, p$, generar z_j^{i+1} de $Z_j \mid z_1^{i+1}, \dots, z_{j-1}^{i+1}, z_{j+1}^i, \dots, z_p^i \sim N_{S_j} = (\alpha_j, 1)$.
3. Obtenga $\boldsymbol{\beta}^{i+1} = \mathbf{A}^{-1}\mathbf{z}^{i+1}$, donde $\mathbf{z}^{i+1} = [z_1^{i+1}, \dots, z_p^{i+1}]^t$.
4. Incrementar en 1 a i e ir al paso 2.

De esta manera se obtiene la cadena de Markov $\boldsymbol{\beta}^0, \boldsymbol{\beta}^1, \dots, \boldsymbol{\beta}^r$ la cual converge a la distribución $N_{\mathbf{T}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ para una r apropiada. Para más detalle sobre este algoritmo ver Rodríguez-Yam (2003).

Para la realización del Paso 2, es decir, para muestrear las distribuciones univariadas normales truncadas definidas en (15), se puede utilizar el **Resultado 3.9.4** que se dará en la Sección 3.9.

3.7. Muestreo de Importancia (IS)

El muestreo de importancia sugerido por Fosdick (1963), corrige la estimación de $E_{\pi}[h(\mathbf{x})]$ con respecto a una distribución estacionaria $\pi(\cdot)$ de una cadena de Markov; basándose en una muestra de una aproximación de $\pi(\cdot)$.

Más específicamente, sea $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$ una cadena de Markov con distribución estacionaria $\pi(\cdot)$ y $h(\mathbf{x})$ una función real del vector \mathbf{x} . Supóngase que se está interesado en calcular $E_{\pi}[h(\mathbf{x})]$ con respecto a la distribución estacionaria $\pi(\cdot)$. Sea $\pi^*(\cdot)$ una aproximación de $\pi(\cdot)$ y sea $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$

una muestra de $\pi^*(\cdot)$. En la Técnica IS, cada punto de la muestra es *ponderado* para corregir las probabilidades de muestreo de que la muestra ponderada pueda estar relacionada con la distribución $\pi(\cdot)$. Los pesos usados para corregir las probabilidades son

$$\omega_i := \frac{\pi(\mathbf{x}_i)/\pi^*(\mathbf{x}_i)}{\sum_{i=1}^n \pi(\mathbf{x}_i)/\pi^*(\mathbf{x}_i)}, \quad i = 1, \dots, n. \quad (17)$$

Entonces, $E_\pi[h(\mathbf{x})]$ puede calcularse como:

$$E_\pi[h(\mathbf{x})] = \frac{\int h(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}}{\int \pi(\mathbf{x})d\mathbf{x}} = \frac{\int h(\mathbf{x})\frac{\pi(\mathbf{x})}{\pi^*(\mathbf{x})}\pi^*(\mathbf{x})d\mathbf{x}}{\int \frac{\pi(\mathbf{x})}{\pi^*(\mathbf{x})}\pi^*(\mathbf{x})d\mathbf{x}}. \quad (18)$$

Así, la ecuación (18) sugiere que una aproximación MCMC para estimar $E_\pi[h(\mathbf{x})]$ es a través del estimador

$$\hat{E}_\pi[h(\mathbf{x})] = \sum_{i=1}^n \omega_i h(\mathbf{x}_i), \quad (19)$$

donde ω_i está dado en (17).

En general, si $\pi^*(\cdot)$ difiere sustancialmente de $\pi(\cdot)$, $\hat{E}_\pi[h(\mathbf{x})]$ es inestable y su varianza es grande, lo cual podría hacer más larga la corrida de la cadena de Markov. Es importante que $\pi(\cdot)/\pi^*(\cdot)$ esté acotada y que $\pi^*(\cdot)$ tenga colas más pesadas que $\pi(\cdot)$. Una característica atractiva de la técnica IS es la posibilidad de reusar las simulaciones; los mismos puntos muestreados y ponderados pueden ser usados para calcular una variedad de estimaciones MCMC de diferentes cantidades. Las ponderaciones pueden ser cambiadas para considerar otra aproximación a la distribución $\pi(\cdot)$, para evaluar o mejorar el comportamiento del estimador por sí mismo o para considerar una distribución alternativa $\pi(\cdot)$, y de este modo poder estimar la esperanza de $h(\mathbf{x})$ con respecto a una distribución diferente. Para más detalles ver Givens y Hoeting (2005).

3.8. Aproximación normal de la verosimilitud

Sea $l(\theta)$ el logaritmo de la función de verosimilitud de un parámetro continuo θ con espacio paramétrico Ω ; sean $S(\theta) = l'(\theta)$ y $\zeta(\theta) = -l''(\theta)$ las funciones Score y de Información respectivamente. Supóngase que $l(\theta)$ tiene una expansión en serie de Taylor en $\hat{\theta} \in \Omega$, i.e.,

$$l(\theta) - l(\hat{\theta}) = (\theta - \hat{\theta})l'(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^2 l''(\hat{\theta}) + \frac{1}{3!}(\theta - \hat{\theta})^3 l'''(\hat{\theta}) + \dots$$

La aproximación normal a $r(\theta) = l(\theta) - l(\hat{\theta})$ se define como

$$r_N(\theta) = -\frac{1}{2}(\theta - \hat{\theta})^2 \zeta(\theta).$$

Si $|\theta - \hat{\theta}|$ es pequeño, entonces los términos cúbicos y de más potencia son más pequeños, por lo que $r(\theta) \approx r_N(\theta)$. El efecto de incrementar la muestra da como resultado intervalos de verosimilitud más estrechos para θ . Así, para muestras suficientemente grandes, $|\theta - \hat{\theta}|$ será pequeño y $r_N(\theta)$ proporcionará una buena aproximación para $r(\theta)$ en todo el espacio Ω .

Se puede checar la exactitud de la aproximación normal en cada situación, graficando $r(\theta)$ y $r_N(\theta)$ en la misma gráfica y verificando la cercanía de ambas funciones en un intervalo de θ . Para más detalles ver Kalbfleisch, J. G. (1985).

3.9. Otros resultados

En esta Sección se muestran cuatro resultados para la implementación de la metodología propuesta en este trabajo. En el primer resultado se proporciona una condición de existencia para los puntos óptimos de una función. El segundo proporciona una identidad matricial para la suma de dos formas cuadráticas. El tercero proporciona una identidad para una clase especial de forma cuadrática y el último proporciona una manera de encontrar un valor aleatorio de una variable truncada en un intervalo.

Resultado 3.9.1 Sea $f : \Re^n \rightarrow \Re$ una función continua en un conjunto compacto \mathbf{C} de \Re^n , entonces existen puntos \mathbf{c} y \mathbf{d} en \mathbf{C} tales que:

$$f(\mathbf{c}) = \sup f \quad y \quad f(\mathbf{d}) = \inf f$$

Este resultado conocido como el Teorema del valor extremo, puede encontrarse en cualquier libro de cálculo avanzado, como por ejemplo Apostol (1969).

Resultado 3.9.2 Sean \mathbf{z} , \mathbf{a} y \mathbf{b} , vectores de orden k y \mathbf{A} y \mathbf{B} matrices simétricas de orden $k \times k$ tal que $(\mathbf{A} + \mathbf{B})^{-1}$ existe, entonces:

$$\begin{aligned} (\mathbf{z} - \mathbf{a})^t \mathbf{A}(\mathbf{z} - \mathbf{a}) + (\mathbf{z} - \mathbf{b})^t \mathbf{B}(\mathbf{z} - \mathbf{b}) &= (\mathbf{z} - \mathbf{c})^t (\mathbf{A} + \mathbf{B})(\mathbf{z} - \mathbf{c}) \\ &\quad + (\mathbf{a} - \mathbf{b})^t \mathbf{A}(\mathbf{A} + \mathbf{B})^{-1} \mathbf{B}(\mathbf{a} - \mathbf{b}) \end{aligned}$$

donde $\mathbf{c} = (\mathbf{A} + \mathbf{B})^{-1}(\mathbf{A}\mathbf{a} + \mathbf{B}\mathbf{b})$. Para una prueba de esta resultado ver Box y Tiao (1973).

Resultado 3.9.3 Sean \mathbf{w} y $\boldsymbol{\beta}$ vectores de orden n y p , respectivamente y \mathbf{X} una matriz de orden $n \times p$, entonces

$$(\mathbf{w} - \mathbf{X}\boldsymbol{\beta})^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} (\mathbf{w} - \mathbf{X}\boldsymbol{\beta}) = s(\mathbf{w}) + (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^t \mathbf{X}^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \mathbf{X} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}),$$

donde

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \mathbf{X})^{-1} \mathbf{X}^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \mathbf{w}, \quad (20)$$

$$s(\mathbf{w}) = (\mathbf{w} - \mathbf{X}\hat{\boldsymbol{\beta}})^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} (\mathbf{w} - \mathbf{X}\hat{\boldsymbol{\beta}}). \quad (21)$$

Resultado 3.9.4 Sea X una variable aleatoria con función de densidad y distribución acumulada, $f(\cdot)$ y $F(\cdot)$ respectivamente, entonces la densidad de X truncada en el intervalo $[a, b]$ está dada por

$$g(x) = \frac{f(x)I_{[a,b]}(x)}{F(b) - F(a)}, \quad -\infty < a < b < \infty.$$

Además, si u es una variable aleatoria con distribución uniforme en $(0, 1)$, entonces una realización x de g está dada por

$$x = F^{-1}(F(a) + u(F(b) - F(a)))$$

Devroye (1986).

4. MATERIALES Y MÉTODOS

Se considera un MLG con parámetro de escala conocido, cuyo vector de coeficientes de regresión se encuentra sujeto a un conjunto de *restricciones lineales de desigualdades* que definen una región que tiene medida de Lebesgue positiva. El número de restricciones lineales puede exceder la dimensión del vector de coeficientes de regresión.

En esta Sección se plantea un modelo Bayesiano para el vector de coeficientes de regresión y se proporciona un método fácil de implementar denominado Gibbs Transformado con Muestreo de Importancia (GTMI). La idea principal de este método consiste en aproximar la verosimilitud del MLG alrededor de la moda de la distribución final, con una distribución normal multivariada truncada en el vector de coeficientes de regresión, la cual se muestrea eficientemente con el MGRY. Para realizar inferencias sobre el vector de coeficientes de regresión se utiliza la técnica IS. Para la obtención de la moda del modelo Bayesiano se utiliza metodología de regresión lineal ponderada. Para el análisis Bayesiano, se consideran los casos cuando la distribución inicial para el vector de coeficientes de regresión es una Uniforme y cuando es Normal multivariada truncada.

Se proporcionan condiciones de regularidad bajo las cuales el GTMI puede implementarse.

A través de un estudio de simulación se comparan los resultados obtenidos por el GTMI y por el método propuesto por Dellaportas y Smith (1993), denominado GDS. Para este estudio se considera un MLG Bernoulli en el cual el vector de coeficientes de regresión está sujeto a un conjunto de restricciones lineales; por cuestiones ilustrativas la dimensión del vector de coeficientes de regresión es dos y se consideran tres casos para el verdadero valor del vector de coeficientes de regresión. Se monitorea la convergencia de ambos métodos, basada en tres cadenas de Markov independientes con longitud 10,000 y puntos de inicio dispersos a través del espacio restringido. Para disminuir el efecto del punto de inicio, como regla práctica, se descarta la primera mitad de cada cadena de Markov y para realizar inferencias se mezclan las segundas mitades. Se calcula la variación entre y dentro de cada cadena y se obtiene el factor de reducción de escala potencial para cada componente del vector de coeficientes de regresión. Se estima la media, desviación estándar y se obtiene el sesgo para cada componente del vector de coeficientes de regresión. Para evaluar el impacto de la aproximación normal a la verosimilitud, se consideran tamaños de muestras 30 y 100.

Para evaluar el desempeño del GTMI en dimensiones mayores a dos, se realiza un estudio de simulación para el MLG exponencial con cinco componentes para el vector de coeficientes de regresión, dos casos para el verdadero valor del vector de coeficientes de regresión y tamaños de muestras 30, 100 y 400.

En Ambos estudios se considera una distribución apriori no informativa para el análisis Bayesiano.

4.1. Planteamiento del modelo bayesiano

En esta Sección se plantea un modelo Bayesiano al vector de coeficientes de regresión de un MLG con parámetro de escala conocido, el cual se encuentra sujeto a restricciones de desigualdad.

Para la componente aleatoria del MLG se hacen las siguientes consideraciones: Sea $\mathbf{y} := [y_1, \dots, y_n]^t$ el vector de respuestas cuyo vector de media es $\boldsymbol{\mu} := (\mu_1, \dots, \mu_n)^t$ y sea \mathbf{X} la matriz de diseño de rango p y de orden $n \times p$ ($n > p$). Las componentes y_i del vector \mathbf{y} son independientes y su distribución pertenece a la familia exponencial. Suponiendo que el parámetro de escala es conocido, entonces por (3) y (5) la función de densidad de y_i , $i = 1, \dots, n$, es de la forma:

$$\begin{aligned} f(y_i; \theta_i) &\propto \exp[y_i b(\theta_i) + c(\theta_i)] \\ &= \exp[y_i b(h(\mathbf{x}_i^t \boldsymbol{\beta})) + c(h(\mathbf{x}_i^t \boldsymbol{\beta}))], \quad i = 1, \dots, n \end{aligned} \quad (22)$$

Para la componente sistemática del MLG, se considera:

$$\eta_i = g(\mu_i), \quad i = 1, \dots, n, \quad (23)$$

donde η_i es el predictor lineal tal que $\eta_i = \mathbf{x}_i^t \boldsymbol{\beta}$ y $\boldsymbol{\beta} := (\beta_1, \dots, \beta_p)^t$ es el vector de coeficientes de regresión que relaciona a $g(\mu_i)$ con \mathbf{x}_i^t . Suponiendo que el vector $\boldsymbol{\beta}$ está sujeto a un conjunto de restricciones lineales de desigualdad dado por

$$\mathbf{B}\boldsymbol{\beta} \leq \mathbf{b}, \quad (24)$$

donde \mathbf{B} es una matriz de constantes conocidas de orden $q \times p$ y \mathbf{b} un vector de constantes conocidas de orden q . Las restricciones lineales en (24) definen el siguiente subespacio paramétrico convexo para el vector $\boldsymbol{\beta}$:

$$\mathbf{T} := \{\boldsymbol{\beta} \in \mathbb{R}^p \mid \mathbf{B}\boldsymbol{\beta} \leq \mathbf{b}\}. \quad (25)$$

Supóngase que el conjunto \mathbf{T} tiene medida de Lebesgue positiva. La verosimilitud $L(\boldsymbol{\theta}; \mathbf{y})$ del vector \mathbf{y} , definida por el MLG obtenido por la componente aleatoria en (22) y la componente sistemática en (23), está dada por:

$$\begin{aligned} L(\boldsymbol{\theta}; \mathbf{y}) &\propto \exp \left[\sum_{i=1}^n [y_i b(h(\mathbf{x}_i^t \boldsymbol{\beta})) + c(h(\mathbf{x}_i^t \boldsymbol{\beta}))] \right] I_{\mathbf{T}}(\boldsymbol{\beta}) \\ &= L(\boldsymbol{\beta}; \mathbf{y}) I_{\mathbf{T}}(\boldsymbol{\beta}), \end{aligned} \quad (26)$$

donde $I_{\mathbf{T}}(\cdot)$ es la función indicadora, i.e., $I_{\mathbf{T}}(\boldsymbol{\beta}) = 1$ si $\boldsymbol{\beta} \in \mathbf{T}$ y 0 de otro modo.

Sea $\pi(\boldsymbol{\beta})$ una distribución inicial del vector $\boldsymbol{\beta}$ sujeta a la restricción $\boldsymbol{\beta} \in \mathbf{T}$. Considerando la función de verosimilitud en (26) y la distribución inicial $\pi(\boldsymbol{\beta})$, el teorema de Bayes implica que la distribución final del vector $\boldsymbol{\beta}$, sujeta a la restricción $\boldsymbol{\beta} \in \mathbf{T}$, esta determinada por

$$\begin{aligned} \pi(\boldsymbol{\beta}|\mathbf{y}) &= \frac{L(\boldsymbol{\beta}; \mathbf{y})\pi(\boldsymbol{\beta})I_{\mathbf{T}}(\boldsymbol{\beta})}{\int_{\mathbf{T}} L(\boldsymbol{\beta}; \mathbf{y})\pi(\boldsymbol{\beta})\partial\boldsymbol{\beta}} \propto L(\boldsymbol{\beta}; \mathbf{y})\pi(\boldsymbol{\beta})I_{\mathbf{T}}(\boldsymbol{\beta}), \\ &= \exp \left[\sum_{i=1}^n [y_i b(h(\mathbf{x}_i^t; \boldsymbol{\beta})) + c(h(\mathbf{x}_i^t; \boldsymbol{\beta}))] \right] \pi(\boldsymbol{\beta})I_{\mathbf{T}}(\boldsymbol{\beta}). \end{aligned} \quad (27)$$

Como resultado de la restricción impuesta por el conjunto \mathbf{T} definido en (25), la evaluación analítica de la constante de normalización es típicamente intratable. Además, cuando la región de truncamiento tiene una probabilidad pequeña con respecto a la distribución no restringida, el cálculo y el análisis Bayesiano se hace extremadamente difícil, por lo que la implementación directa de los métodos de simulación Monte Carlo (MCMC), el muestreo Gibbs (Geman y Geman, 1984; Gelfand y Smith, 1990) y el Metropolis-Hastings (Metropolis *et al.*, 1953; Hastings, 1970), resultan ser muy ineficientes.

4.2. Gibbs transformado con muestreo de importancia (GTMI)

En esta Sección se propone una metodología fácil de implementar, la cual permite realizar inferencias sobre el vector $\boldsymbol{\beta}$ a través de la estimación de $E[h(\boldsymbol{\beta})]$, donde $h(\boldsymbol{\beta})$ es una función real del vector $\boldsymbol{\beta}$ y la esperanza se calcula con respecto a la distribución en (27).

La idea principal de esta metodología consiste en aproximar la verosimilitud $L(\boldsymbol{\beta}; \mathbf{y})$ en (26), con una distribución normal, de tal manera que bajo ciertas condiciones de regularidad, la verosimilitud $L(\boldsymbol{\beta}; \mathbf{y})$ se aproxime a la verosimilitud de un modelo con distribución normal, y de esta forma, se obtenga una aproximación $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ de la distribución en (27). Bajo estas condiciones, se implementa el MGRY para muestrear eficientemente la distribución $\pi_a(\boldsymbol{\beta}|\mathbf{y})$.

4.2.1. Aproximación normal de $L(\boldsymbol{\beta}; \mathbf{y})$

Sea $\hat{\boldsymbol{\beta}}$ una estimación puntual de la moda de la distribución final en (27), donde las componentes del vector $\mathbf{y} := (y_1, \dots, y_n)^t$ son realizaciones independientes del modelo definido en (22). Como función de $\boldsymbol{\beta}$ la verosimilitud en (26) es

$$L(\boldsymbol{\beta}; \mathbf{y}) = \prod_{i=1}^n f(y_i; \eta_i) = \prod_{i=1}^n \exp(\log f(y_i; \eta_i)) = \prod_{i=1}^n \exp(l(\eta_i; y_i)), \quad (28)$$

donde $l(\eta_i; y_i) = \log f(y_i; \eta_i)$. Sean $l'(\hat{\eta}_i; y_i) = [\partial/\partial\eta_i l(\eta_i; y_i)]_{\eta_i=\hat{\eta}_i}$ y $l''(\hat{\eta}_i; y_i) = [\partial^2/\partial^2\eta_i l(\eta_i; y_i)]_{\eta_i=\hat{\eta}_i}$, donde $\hat{\eta}_i = \mathbf{x}_i^t \hat{\boldsymbol{\beta}}$. Desarrollando $l(\eta_i; y_i)$ en serie de Taylor de segundo orden se obtiene que

$$l(\eta_i; y_i) \approx l(\hat{\eta}_i; y_i) + (\eta_i - \hat{\eta}_i)l'(\hat{\eta}_i; y_i) + \frac{1}{2}(\eta_i - \hat{\eta}_i)^2 l''(\hat{\eta}_i; y_i). \quad (29)$$

De donde

$$\begin{aligned} l(\eta_i; y_i) - l(\hat{\eta}_i; y_i) &\approx \frac{1}{2} l''(\hat{\eta}_i; y_i) \left[(\eta_i - \hat{\eta}_i)^2 + 2(\eta_i - \hat{\eta}_i) \frac{l'(\hat{\eta}_i; y_i)}{l''(\hat{\eta}_i; y_i)} \right] \\ &= \frac{1}{2} l''(\hat{\eta}_i; y_i) \left[\eta_i - \left(\hat{\eta}_i - \frac{l'(\hat{\eta}_i; y_i)}{l''(\hat{\eta}_i; y_i)} \right) \right]^2 \\ &= -\frac{1}{2\sigma_i^2} (\eta_i - w_i)^2, \end{aligned}$$

donde

$$w_i = \hat{\eta}_i - \frac{l'(\hat{\eta}_i; y_i)}{l''(\hat{\eta}_i; y_i)}, \quad (30)$$

$$\sigma_i^2 = -\frac{1}{l''(\hat{\eta}_i; y_i)}. \quad (31)$$

Como $l(\hat{\eta}_i; y_i)$ es una constante, entonces

$$f(y_i; \eta_i) \propto \exp \left\{ -\frac{1}{2\sigma_i^2} (\eta_i - w_i)^2 \right\}. \quad (32)$$

Sustituyendo (32) en (28) se obtiene

$$\begin{aligned} L(\boldsymbol{\beta}; \mathbf{y}) &\approx \exp \left\{ -\sum_{i=1}^n \frac{1}{2\sigma_i^2} (w_i - \mathbf{x}_i^t \boldsymbol{\beta})^2 \right\} I_{\mathbf{T}}(\boldsymbol{\beta}) \\ &= \exp \left\{ -\frac{1}{2} (\mathbf{w} - \mathbf{X}\boldsymbol{\beta})^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} (\mathbf{w} - \mathbf{X}\boldsymbol{\beta}) \right\} I_{\mathbf{T}}(\boldsymbol{\beta}) = L(\boldsymbol{\beta}; \mathbf{w}), \end{aligned} \quad (33)$$

donde $\mathbf{w} =: (w_1, \dots, w_n)^t$, $\boldsymbol{\Sigma}_{\mathbf{w}} =: \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$; w_i y σ_i^2 se obtienen de (30) y (31) respectivamente, y éstas dependen de y_i y $\hat{\boldsymbol{\beta}}$. Note que la matriz $\boldsymbol{\Sigma}_{\mathbf{w}}$ es simétrica y positiva definida, ya que $l''(\hat{\eta}_i; y_i) < 0$ implica que $\sigma_i^2 > 0$ para toda $i = 1, \dots, n$.

Por lo tanto, la verosimilitud correspondiente al vector \mathbf{y} , es *aproximadamente* la verosimilitud del vector \mathbf{w} , cuyas componentes son independientes, con matriz de regresores \mathbf{X} y procedente de un *modelo Normal truncado* con vector de medias $\boldsymbol{\mu}_{\mathbf{w}} =: (\mathbf{X}_1^t \boldsymbol{\beta}, \dots, \mathbf{X}_n^t \boldsymbol{\beta})^t$ y matriz de varianzas y covarianzas conocidas $\boldsymbol{\Sigma}_{\mathbf{w}} = \text{diag}[\sigma_1^2, \dots, \sigma_n^2]$.

Estas condiciones corresponden a un *modelo de regresión lineal ponderado*, con restricciones lineales en el vector de coeficientes de regresión, y matriz de varianzas y covarianzas conocida.

4.2.2. Planteamiento aproximado para el modelo Bayesiano $\pi(\boldsymbol{\beta}|\mathbf{y})$

Tomando en cuenta la aproximación de la verosimilitud en (33), se obtiene la aproximación $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ para la distribución en (27)

$$\pi_a(\boldsymbol{\beta}|\mathbf{y}) \propto L(\boldsymbol{\beta}; \mathbf{w})\pi(\boldsymbol{\beta})I_{\mathbf{T}}(\boldsymbol{\beta}) = \exp \left\{ -\frac{1}{2}(\mathbf{w} - \mathbf{X}\boldsymbol{\beta})^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1}(\mathbf{w} - \mathbf{X}\boldsymbol{\beta}) \right\} \pi(\boldsymbol{\beta})I_{\mathbf{T}}(\boldsymbol{\beta}). \quad (34)$$

Para el análisis Bayesiano del modelo $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ en (34), se consideran distribuciones iniciales uniformes y distribuciones iniciales normales multivariadas truncadas, definidas en la región \mathbf{T} en (25).

4.2.2.1. Distribución inicial-Uniforme

Para que la distribución final del vector $\boldsymbol{\beta}$ sea propia cuando se asume una inicial impropia una condición suficiente es que $n > p$ y que el rango de \mathbf{X} sea p (ver Gelman, *et al.*, 2004). Ya que en el planteamiento del modelo se cumplen estas condiciones, puede usarse la siguiente distribución inicial impropia

$$\pi(\boldsymbol{\beta}) \propto \mathbf{1}_{\mathbf{T}}(\boldsymbol{\beta}). \quad (35)$$

De donde la distribución final en (34) es

$$\pi_a(\boldsymbol{\beta}|\mathbf{y}) \propto \exp \left\{ -\frac{1}{2}(\mathbf{w} - \mathbf{X}\boldsymbol{\beta})^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1}(\mathbf{w} - \mathbf{X}\boldsymbol{\beta}) \right\} I_{\mathbf{T}}(\boldsymbol{\beta}). \quad (36)$$

Aplicando el **Resultado 3.9.3** a la expresión $(\mathbf{w} - \mathbf{X}\boldsymbol{\beta})^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1}(\mathbf{w} - \mathbf{X}\boldsymbol{\beta})$, y tomando en cuenta que $s(\mathbf{w})$ en (21) es una constante, entonces la distribución en (36) se puede escribir como

$$\pi_a(\boldsymbol{\beta}|\mathbf{y}) \propto \exp \left\{ -\frac{1}{2}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^t \mathbf{X}^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \mathbf{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \right\} I_{\mathbf{T}}(\boldsymbol{\beta}) \quad (37)$$

Es decir, $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ es una normal multivariada truncada en \mathbf{T} con vector de medias $\hat{\boldsymbol{\beta}}$ dada en (20) y matriz de varianzas y covarianzas $[\mathbf{X}^t \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \mathbf{X}]^{-1}$.

4.2.2.2. Distribución inicial-Normal multivariada

Si se asume que la distribución inicial para el vector $\boldsymbol{\beta}$ es una normal multivariada truncada en \mathbf{T} , es decir,

$$\boldsymbol{\beta} \sim N_{\mathbf{T}}(\boldsymbol{\beta}; \boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0) \quad (38)$$

donde $\boldsymbol{\beta}_0$ es un vector conocido que pertenece al conjunto \mathbf{T} en (25) y $\boldsymbol{\Sigma}_0$ es conocida y positiva definida, por (38) y (34), se tiene que la distribución aproximada del vector $\boldsymbol{\beta}$ sujeta a la restricción $\boldsymbol{\beta} \in \mathbf{T}$, está dada por

$$\begin{aligned} \pi_a(\boldsymbol{\beta}|\mathbf{y}) &\propto \exp \left[-\frac{1}{2} [(\mathbf{w} - \mathbf{X}\boldsymbol{\beta})^t \boldsymbol{\Sigma}_w^{-1} (\mathbf{w} - \mathbf{X}\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^t \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)] \right] I_{\mathbf{T}}(\boldsymbol{\beta}) \\ &= \exp \left[-\frac{1}{2} [(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^t \mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{X} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^t \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)] \right] I_{\mathbf{T}}(\boldsymbol{\beta}) \\ &= \exp \left[-\frac{1}{2} (\boldsymbol{\beta} - \mathbf{c})^t (\mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{X} + \boldsymbol{\Sigma}_0^{-1}) (\boldsymbol{\beta} - \mathbf{c}) \right] I_{\mathbf{T}}(\boldsymbol{\beta}) \end{aligned} \quad (39)$$

donde $\mathbf{c} = (\mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{X} + \boldsymbol{\Sigma}_0^{-1})^{-1} (\mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{w} + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\beta}_0)$. La segunda expresión en (39) se obtiene al aplicar el **Resultado 3.9.3** a la expresión $(\mathbf{w} - \mathbf{X}\boldsymbol{\beta})^t \boldsymbol{\Sigma}_w^{-1} (\mathbf{w} - \mathbf{X}\boldsymbol{\beta})$ y tomando en cuenta que $s(\mathbf{w})$ en (21) es una constante, y la tercera expresión en (39) se obtiene al aplicar el **Resultado 3.9.2** a la expresión $(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^t \mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{X} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^t \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)$. Por lo tanto, $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ en (39) es una normal multivariada truncada en la región \mathbf{T} en (25) con vector de medias

$$\tilde{\boldsymbol{\beta}} = \boldsymbol{\Sigma} (\mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{w} + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\beta}_0), \quad (40)$$

y matriz de varianzas y covarianzas

$$\boldsymbol{\Sigma} = (\mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{X} + \boldsymbol{\Sigma}_0^{-1})^{-1}. \quad (41)$$

4.2.3. Muestreo de la distribución $\pi_a(\boldsymbol{\beta}|\mathbf{y})$

Para obtener una muestra $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_n$ de la distribución $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ dada por (37) o por (39), se implementa el MGRY.

4.2.4. Inferencias para funciones del vector $\boldsymbol{\beta}$

Supóngase que se está interesado en calcular $E_{\pi}[h(\boldsymbol{\beta})]$ con respecto a $\pi(\boldsymbol{\beta}|\mathbf{y})$, donde $h(\boldsymbol{\beta})$ es una función real del vector $\boldsymbol{\beta}$. Sea $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_n$, una muestra de $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ obtenida con la implementación del MGRY. Para calcular $E_{\pi}[h(\boldsymbol{\beta})]$ con respecto a $\pi(\boldsymbol{\beta}|\mathbf{y})$, se aplica la técnica IS utilizando $\pi_a(\boldsymbol{\beta}|\mathbf{y})$.

Así, por (17) y (19) se tiene que

$$\hat{E}_\pi[h(\boldsymbol{\beta})] = \sum_{i=1}^n \omega_i h(\boldsymbol{\beta}_i)$$

donde

$$\omega_i = \frac{\pi(\boldsymbol{\beta}_i|\mathbf{y})/\pi_a(\boldsymbol{\beta}_i|\mathbf{y})}{\sum_{i=1}^n \pi(\boldsymbol{\beta}_i|\mathbf{y})/\pi_a(\boldsymbol{\beta}_i|\mathbf{y})}, \quad i = 1, \dots, n$$

Para la obtención de $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_n$, implementar el siguiente algoritmo

1. Dar un valor inicial $\hat{\boldsymbol{\beta}}_0$ para la moda de $\pi(\boldsymbol{\beta}|\mathbf{y})$.
2. Si la distribución inicial es (35), calcular la moda $\hat{\boldsymbol{\beta}}$ utilizando (20) y calcular $[\mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{X}]^{-1}$, ambos en el punto $\hat{\boldsymbol{\beta}}_0$. Si la distribución inicial es (38), calcular la moda $\tilde{\boldsymbol{\beta}}$ utilizando (40) y calcular $\boldsymbol{\Sigma}$ utilizando (41), ambos en el punto $\hat{\boldsymbol{\beta}}_0$. La matriz $\boldsymbol{\Sigma}_w = \text{diag}[\sigma_1^2, \dots, \sigma_n^2]$, w_i , y σ_i^2 se obtienen de (30) y (31) respectivamente.
3. Hacer $\hat{\boldsymbol{\beta}}_0 = \hat{\boldsymbol{\beta}}$ si la distribución inicial es (35). Hacer $\hat{\boldsymbol{\beta}}_0 = \tilde{\boldsymbol{\beta}}$ si la distribución inicial es (38).
4. Repetir los Pasos 2 y 3 si $\sqrt{(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_0)^t (\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_0) / \hat{\boldsymbol{\beta}}_0^t \hat{\boldsymbol{\beta}}_0} \geq e$, donde e es el error permitido (por ejemplo $e = 0.0001$). De otro modo ir al paso 5.
5. Si la distribución inicial es (35) y si $\hat{\boldsymbol{\beta}}$ y $[\mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{X}]^{-1}$ son los valores obtenidos en la última iteración del Paso 4, entonces $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ es una distribución $N_{\mathbf{T}}(\hat{\boldsymbol{\beta}}, [\mathbf{X}^t \boldsymbol{\Sigma}_w^{-1} \mathbf{X}]^{-1})$. Si la distribución inicial es (38) y si $\tilde{\boldsymbol{\beta}}$ y $\boldsymbol{\Sigma}$ son los valores obtenidos en la última iteración del Paso 4, entonces $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ es una distribución $N_{\mathbf{T}}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\Sigma})$.
6. Implementar el MGRY para obtener una muestra $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_n$ de la distribución $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ obtenida en el paso 5.

Observaciones:

- a) La repetición de los Pasos 2 y 3 es equivalente a un proceso iterativo de una regresión lineal ponderada de las variables dependientes ajustadas w_i sobre las variables explicativas \mathbf{x}_i . Este proceso iterativo equivale a resolver el sistema de ecuaciones $\partial \text{Log} \pi(\boldsymbol{\beta}|\mathbf{y}) / \partial \boldsymbol{\beta} = 0$, mediante el algoritmo “*Score de Fisher*”. Dicho proceso converge a la moda rápidamente para modelos lineales generalizados con funciones ligas canónicas. La justificación de este resultado se hará en la siguiente Sección.
- b) Para calcular la moda o para evaluar la convergencia del GTMI, es necesario partir de puntos que pertenecen al conjunto \mathbf{T} . Para elegir los puntos, se muestrea la distribución

normal truncada $N_{\mathbf{T}}(\mathbf{0}, \sigma^2 \mathbf{I})$ a través del MGRY, a σ^2 se le da un valor grande para cubrir el soporte \mathbf{T} . El punto de inicio para correr el muestreo Gibbs se obtiene de $\mathbf{B}^{-} \mathbf{b}$, donde \mathbf{B}^{-} es la inversa generalizada de la matriz \mathbf{B} , donde la matriz \mathbf{B} y el vector \mathbf{b} están dados en (24).

4.2.5. Condiciones de regularidad

Para implementar el GTMI, se debe aproximar la función de verosimilitud con una distribución normal alrededor de la moda de la distribución final. Es necesario que se cumplan las siguientes condiciones

1. Que exista la moda $\hat{\boldsymbol{\beta}}$ de la distribución $\pi(\boldsymbol{\beta}|\mathbf{y})$.
2. Que $L(\boldsymbol{\beta}; \mathbf{y})$ pueda ser aproximada por una distribución normal multivariada, alrededor de $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$.

4.2.5.1. Para la existencia y obtención de la moda

La moda $\hat{\boldsymbol{\beta}}$ es un valor en el espacio paramétrico de $\boldsymbol{\beta}$ que no necesariamente está en el conjunto \mathbf{T} de (25) y es solución de la ecuación

$$\frac{\partial}{\partial \boldsymbol{\beta}} \log[\pi(\boldsymbol{\beta}|\mathbf{y})] = 0. \quad (42)$$

Tomando en cuenta la Definición de $\pi(\boldsymbol{\beta}|\mathbf{y})$ en (27), se tiene que

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\beta}} \log[\pi(\boldsymbol{\beta}|\mathbf{y})] &\propto \frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^n [y_i h(\mathbf{x}_i^t; \boldsymbol{\beta}) - b(h(\mathbf{x}_i^t; \boldsymbol{\beta}))] + \frac{\partial}{\partial \boldsymbol{\beta}} \log \pi(\boldsymbol{\beta}) \\ &= \sum_{i=1}^n [y_i - b'(h(\mathbf{x}_i^t; \boldsymbol{\beta}))] h'(\mathbf{x}_i^t; \boldsymbol{\beta}) \mathbf{x}_i + \frac{\partial}{\partial \boldsymbol{\beta}} \log \pi(\boldsymbol{\beta}). \end{aligned} \quad (43)$$

Para encontrar la moda $\hat{\boldsymbol{\beta}}$ se resuelve (42) usando el siguiente paso iterativo del algoritmo “*Score de Fisher*”

$$\hat{\boldsymbol{\beta}}^t = \hat{\boldsymbol{\beta}}^{t-1} - \left\{ E \left\{ \left[\frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \log [\pi(\boldsymbol{\beta}|\mathbf{y})] \right]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{t-1}} \right\} \right\}^{-1} \left[\frac{\partial}{\partial \boldsymbol{\beta}} \log [\pi(\boldsymbol{\beta}|\mathbf{y})] \right]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{t-1}}. \quad (44)$$

Derivando (43) se tiene que

$$\begin{aligned} \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \log [\pi(\boldsymbol{\beta} | \mathbf{y})] &\propto \sum_{i=1}^n \left[[y_i - b'(h(\mathbf{x}_i^t \boldsymbol{\beta}))] h''(\mathbf{x}_i^t \boldsymbol{\beta}) - [h'(\mathbf{x}_i^t \boldsymbol{\beta})]^2 b''(h(\mathbf{x}_i^t \boldsymbol{\beta})) \right] \mathbf{x}_i \mathbf{x}_i^t \\ &+ \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \log \pi(\boldsymbol{\beta}). \end{aligned} \quad (45)$$

Tomando la esperanza de los datos y_i en la ecuación (45) se tiene que

$$E \left\{ \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \log [\pi(\boldsymbol{\beta} | \mathbf{y})] \right\} \propto - \sum_{i=1}^n [h'(\mathbf{x}_i^t \boldsymbol{\beta})]^2 b''(h(\mathbf{x}_i^t \boldsymbol{\beta})) \mathbf{x}_i \mathbf{x}_i^t + \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \log \pi(\boldsymbol{\beta}). \quad (46)$$

Para simplificar la notación, sean las matrices $V(\boldsymbol{\beta}) = \text{diag}[b''(h(\mathbf{x}_1^t \boldsymbol{\beta})), \dots, b''(h(\mathbf{x}_n^t \boldsymbol{\beta}))]$ y $\Delta(\boldsymbol{\beta}) = \text{diag}[h'(\mathbf{x}_1^t \boldsymbol{\beta}), \dots, h'(\mathbf{x}_n^t \boldsymbol{\beta})]$, entonces (46) es equivalente a

$$E \left\{ \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \log [\pi(\boldsymbol{\beta} | \mathbf{y})] \right\} = -\mathbf{X}^t V(\boldsymbol{\beta}) \Delta^2(\boldsymbol{\beta}) \mathbf{X} + \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \log \pi(\boldsymbol{\beta}). \quad (47)$$

Por otro lado, si definimos los vectores $\tilde{\mathbf{y}} = (\tilde{y}_1, \dots, \tilde{y}_n)^t$ y $\mathbf{z} = (z_1, \dots, z_n)^t$, donde

$$\begin{aligned} \tilde{y}_i &= \frac{y_i - b'(h(\mathbf{x}_i^t \boldsymbol{\beta}))}{h'(\mathbf{x}_i^t \boldsymbol{\beta}) b'(h(\mathbf{x}_i^t \boldsymbol{\beta}))}, \quad i = 1, \dots, n, \\ z_i &= \mathbf{x}_i^t \boldsymbol{\beta} + \tilde{y}_i, \quad i = 1, \dots, n, \end{aligned} \quad (48)$$

entonces (43) es equivalente a

$$\frac{\partial}{\partial \boldsymbol{\beta}} \log [\pi(\boldsymbol{\beta} | \mathbf{y})] = \mathbf{X}^t V(\boldsymbol{\beta}) \Delta^2(\boldsymbol{\beta}) \tilde{\mathbf{y}} + \frac{\partial}{\partial \boldsymbol{\beta}} \log \pi(\boldsymbol{\beta}). \quad (49)$$

Sustituyendo (47) y (49) en (44) se tiene que

$$\begin{aligned} \hat{\boldsymbol{\beta}}^t &= \hat{\boldsymbol{\beta}}^{t-1} + \left\{ \mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \mathbf{X} - \left[\frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} \log \pi(\boldsymbol{\beta}) \right]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{t-1}} \right\}^{-1} \\ &\times \left[\mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \tilde{\mathbf{y}} + \left[\frac{\partial}{\partial \boldsymbol{\beta}} \log \pi(\boldsymbol{\beta}) \right]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{t-1}} \right]. \end{aligned} \quad (50)$$

Si $\pi(\boldsymbol{\beta}) \propto \mathbf{1}$, entonces la moda es el estimador de máxima verosimilitud en (50)

$$\begin{aligned} \hat{\boldsymbol{\beta}}^t &= \hat{\boldsymbol{\beta}}^{t-1} + [\mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \mathbf{X}]^{-1} \mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \tilde{\mathbf{y}} \\ &= [\mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \mathbf{X}]^{-1} \mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \mathbf{z}. \end{aligned} \quad (51)$$

En cada paso iterativo del algoritmo “*Score de Fisher*” resulta una regresión lineal ponderada de las variables z_i sobre la variable explicativa \mathbf{x}_i . La variable z_i en (48) es exactamente la misma que la variable w_i en (30), además, $\Sigma_{\mathbf{w}}^{-1} = \text{diag}[1/\sigma_1^2, \dots, 1/\sigma_n^2] = V(\boldsymbol{\beta})\Delta^2(\boldsymbol{\beta})$ donde σ_i^2 está dada en (31), así, la expresión en (51) es equivalente a la expresada en (44).

Si la distribución inicial $\pi(\boldsymbol{\beta})$ es una $\mathbf{N}_{\mathbf{T}}(\boldsymbol{\beta}; \boldsymbol{\beta}_0, \Sigma_0)$ entonces $\partial/\partial\boldsymbol{\beta} \log \pi(\boldsymbol{\beta}) = -\Sigma_0^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)$ y $\partial^2/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t \log \pi(\boldsymbol{\beta}) = -\Sigma_0^{-1}$, por lo que (50) reduce a

$$\begin{aligned}\hat{\boldsymbol{\beta}}^t &= \hat{\boldsymbol{\beta}}^{t-1} + [\mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \mathbf{X} + \Sigma_0^{-1}]^{-1} [\mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \tilde{\mathbf{y}} - \Sigma_0^{-1}(\hat{\boldsymbol{\beta}}^{t-1} - \boldsymbol{\beta}_0)] \\ &= [\mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \mathbf{X} + \Sigma_0^{-1}]^{-1} [\mathbf{X}^t V(\hat{\boldsymbol{\beta}}^{t-1}) \Delta^2(\hat{\boldsymbol{\beta}}^{t-1}) \mathbf{z} + \Sigma_0^{-1} \boldsymbol{\beta}_0].\end{aligned}\quad (52)$$

Análogamente se tiene que (52) es equivalente a (40).

Si el conjunto \mathbf{T} en (25) es compacto (cerrado y acotado) en \Re^p y $\log[\pi(\boldsymbol{\beta}|\mathbf{y})]$ es continua en \mathbf{T} , entonces por el **Resultado 3.9.1** el supremo de $\log[\pi(\boldsymbol{\beta}|\mathbf{y})]$ existe y está contenido en \mathbf{T} , es decir, la ecuación (42) tiene solución y por lo tanto el algoritmo “*Score de Fisher*” converge.

Otra condición suficiente para que siempre exista solución y además única en (42), es que la matriz $\partial^2/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t \log[\pi(\boldsymbol{\beta}|\mathbf{y})]$ en (45) sea negativa definida, es decir, que la distribución $\pi(\boldsymbol{\beta}|\mathbf{y})$ sea log-cóncava. Para el caso cuando la función liga en (23) sea una función canónica, esta condición siempre se tendrá, ya que para este caso, $\partial^2/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t \log[\pi(\boldsymbol{\beta}|\mathbf{y})]$ en (45) será equivalente a $E\{\partial^2/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t \log[\pi(\boldsymbol{\beta}|\mathbf{y})]\}$ en (47). Así, $\partial^2/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t \log[\pi(\boldsymbol{\beta}|\mathbf{y})] = -[\mathbf{X}^t V(\boldsymbol{\beta}) \Delta^2(\boldsymbol{\beta}) \mathbf{X}]$ o $\partial^2/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t \log[\pi(\boldsymbol{\beta}|\mathbf{y})] = -[\mathbf{X}^t V(\boldsymbol{\beta}) \Delta^2(\boldsymbol{\beta}) \mathbf{X} + \Sigma_0^{-1}]$ dependiendo de cual sea la distribución inicial de $\boldsymbol{\beta}$, Uniforme o $\mathbf{N}_{\mathbf{T}}(\boldsymbol{\beta}; \boldsymbol{\beta}_0, \Sigma_0)$. Como la matriz $\mathbf{X}^t V(\boldsymbol{\beta}) \Delta^2(\boldsymbol{\beta}) \mathbf{X}$ siempre es positiva definida y Σ_0 es positiva definida por el supuesto en (38), entonces $\partial^2/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t \log[\pi(\boldsymbol{\beta}|\mathbf{y})]$ será negativa definida en ambos casos y por lo tanto $\pi(\boldsymbol{\beta}|\mathbf{y})$ será log-cóncava. Tanto en el caso cuando la distribución inicial de $\boldsymbol{\beta}$ es Uniforme como cuando es una $\mathbf{N}_{\mathbf{T}}(\boldsymbol{\beta}; \boldsymbol{\beta}_0, \Sigma_0)$, la log-concavidad de $\pi(\boldsymbol{\beta}|\mathbf{y})$ se reduce a la log-concavidad de la función de verosimilitud. En Wedderburn (1976) se proporciona una serie de casos especiales para funciones ligas no canónicas, en las que se prueba la log-concavidad de la función de verosimilitud para diferentes modelos lineales generalizados.

4.2.5.2. Para la aproximación normal

Para que una expansión de Taylor de $l(\eta_i; y_i) = \log f(y_i; \eta_i)$ en (29) sea posible alrededor de $\eta_i = \hat{\eta}_i = \mathbf{x}_i^t \hat{\boldsymbol{\beta}}$, es necesario que $f(y_i; \hat{\eta}_i) \neq 0$ y $l''(\hat{\eta}_i; y_i) < 0$, $i = 1, \dots, n$. Además, para que la expansión se aproxime a una función cuadrática, basta que la muestra sea suficientemente grande (ver Sección 3.8). Si $\pi(\boldsymbol{\beta}|\mathbf{y})$ es aproximadamente normal, entonces la moda se aproxima a la media y $\left\{ - \left[\partial^2/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t \log[\pi(\boldsymbol{\beta}|\mathbf{y})] \right]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} \right\}^{-1}$ a la matriz de covarianzas.

5. RESULTADOS

En esta Sección se realiza un estudio de simulación para un MLG Bernoulli para comparar los métodos GTMI y GDS. También se muestran los resultados de un estudio de simulación donde sólo se implementa el GTMI para un MLG exponencial, la razón es porque se utiliza una función liga que hace que el modelo bayesiano resultante no sea cóncavo, además, en este estudio se incrementa significativamente la dimensión del vector de coeficientes de regresión con respecto al primer estudio de simulación. En Ambos estudios se consideró una distribución apriori no informativa para el análisis Bayesiano.

5.1. Distribución Bernoulli

La función de distribución de las observaciones que se considera en esta Sección es la distribución Bernoulli, i.e.,

$$\begin{aligned} f(y_i) &= \mu_i^{y_i} (1 - \mu_i)^{1-y_i}, \quad y_i = 0, 1, \quad i = 1, \dots, n, \\ &= \exp \left[y_i \log \left(\frac{\mu_i}{1 - \mu_i} \right) + \log(1 - \mu_i) \right]. \end{aligned} \quad (53)$$

Se consideró la liga canónica $g(\mu_i) = \log(\mu_i/(1 - \mu_i))$, con $g(\mu_i) = \eta_i = \mathbf{x}_i^t \boldsymbol{\beta}$. El vector $\boldsymbol{\beta}$ de orden 2 está sujeto a las restricciones $-\beta_1 \leq 0$, $-\beta_2 \leq 0$, $-\beta_1 + \beta_2 \leq 5$, $\beta_1 + 4\beta_2 \leq 40$, $2\beta_1 + \beta_2 \leq 27$, $3\beta_1 - 4\beta_2 \leq 24$. La región \mathbf{T} en (25) para este conjunto de restricciones es

$$\mathbf{T} := \{ \boldsymbol{\beta} = (\beta_1, \beta_2) \in \Re^2 \mid \mathbf{B}\boldsymbol{\beta} \leq \mathbf{b} \}, \quad (54)$$

donde

$$\mathbf{B} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ -1 & 1 \\ 1 & 4 \\ 2 & 1 \\ 3 & -4 \end{bmatrix} \quad \text{y} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 5 \\ 40 \\ 27 \\ 24 \end{bmatrix}.$$

Se consideran los siguientes valores para el vector $\boldsymbol{\beta}$ en la región \mathbf{T} en (54):

$$(4, 3)^t, (4.5, 8.3)^t \text{ y } (6.2, 0.8)^t$$

La región \mathbf{T} en (54) y los valores para el vector $\boldsymbol{\beta}$ se muestran en la Figura 1.

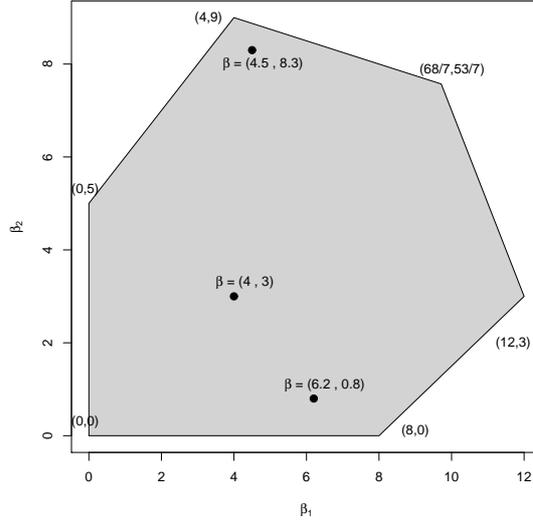


Figura 1: Gráfica de la región \mathbf{T} .

5.1.1. Caso 1: $\beta = (4, 3)^t$

5.1.1.1. Resultados para $n = 30$

Para el caso $n = 30$, se simularon 30 datos de la distribución Bernoulli con probabilidad $\mu_i = 1/(1 + \exp(-\mathbf{x}_i^t \boldsymbol{\beta}))$ donde \mathbf{X} es una matriz de 30×2 . La función de verosimilitud está dada por

$$L(\boldsymbol{\beta}; \mathbf{y}) \propto \exp \left[\sum_{i=1}^{30} y_i \mathbf{x}_i^t \boldsymbol{\beta} - \sum_{i=1}^{30} \log(1 + \exp(\mathbf{x}_i^t \boldsymbol{\beta})) \right] I_{\mathbf{T}}(\boldsymbol{\beta}),$$

y las componentes w_i y σ_i^2 en (30) y (31) son

$$w_i = \hat{\eta}_i + \frac{(1 + e^{\hat{\eta}_i})^2}{e^{\hat{\eta}_i}} \left(\frac{y_i}{\hat{\eta}_i} - \frac{e^{\hat{\eta}_i}}{1 + e^{\hat{\eta}_i}} \right),$$

$$\sigma_i^2 = \frac{1}{\hat{\eta}_i} \frac{(1 + e^{\hat{\eta}_i})^2}{e^{\hat{\eta}_i}}.$$

La distribución inicial para el vector $\boldsymbol{\beta}$ es la no informativa, i.e., $\pi(\boldsymbol{\beta}) = \mathbf{1}_{\mathbf{T}}(\boldsymbol{\beta})$, de donde la distribución final del vector $\boldsymbol{\beta}$ está dada por

$$\pi(\boldsymbol{\beta} | \mathbf{y}) \propto L(\boldsymbol{\beta}; \mathbf{y}) \propto \exp \left[\sum_{i=1}^{30} y_i \mathbf{x}_i^t \boldsymbol{\beta} - \sum_{i=1}^{30} \log(1 + \exp(\mathbf{x}_i^t \boldsymbol{\beta})) \right] I_{\mathbf{T}}(\boldsymbol{\beta}). \quad (55)$$

En el Cuadro 1 se muestran los 30 datos junto con los valores de las dos covariables que se usaron en este modelo.

Cuadro 1: Covariables y realizaciones del modelo en (53).

Num.	\mathbf{y}	\mathbf{x}_1	\mathbf{x}_2
1	1	-0.34	-0.25
2	0	-1.02	0.68
3	0	-1.08	0.31
4	0	-0.46	-0.17
5	0	0.29	-1.83
6	0	-1.01	-0.05
7	1	0.73	2.27
8	0	-0.32	0.39
9	1	0.49	0.59
10	1	0.39	0.85
11	0	-1.49	-1.24
12	0	0.15	-0.86
13	0	-0.45	-1.37
14	1	0.88	-0.05
15	1	1.36	-0.94
16	0	-0.42	-0.51
17	1	0.14	0.20
18	0	-0.50	-0.00
19	0	-0.16	-1.12
20	0	-1.30	0.68
21	1	1.87	0.25
22	1	-0.10	0.67
23	1	0.13	-0.57
24	1	1.28	-0.33
25	0	-1.17	-0.53
26	1	0.67	0.11
27	0	-1.29	-0.94
28	1	-0.32	0.40
29	0	-0.01	-1.06
30	1	0.64	1.03

Para la implementación del GTMI se propuso el punto $(0, 0)^t$ como valor de inicio para la obtención de la moda de la distribución en (55), obteniéndose para el vector $\boldsymbol{\beta}$ la aproximación

$$\pi_a(\boldsymbol{\beta}|\mathbf{y}) = \frac{\sqrt{|\boldsymbol{\Sigma}^{-1}|}}{2\pi} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^t \boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\right\} I_T(\boldsymbol{\beta}), \quad (56)$$

donde la media, la matriz de covarianzas y la matriz de correlaciones están dadas respectivamente por

$$\hat{\boldsymbol{\beta}} = \begin{pmatrix} 4.51 \\ 2.51 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 3.50 & 0.82 \\ 0.82 & 1.70 \end{pmatrix}, \quad \boldsymbol{\rho} = \begin{pmatrix} 1 & 0.33 \\ 0.33 & 1 \end{pmatrix}.$$

En la Figura 2 se muestran las gráficas de contornos correspondiente a las distribuciones en (55) y (56). En esta figura se observa que aproximadamente el 95% de los valores correspondiente a β se encuentran dentro de la misma región en ambas distribuciones.

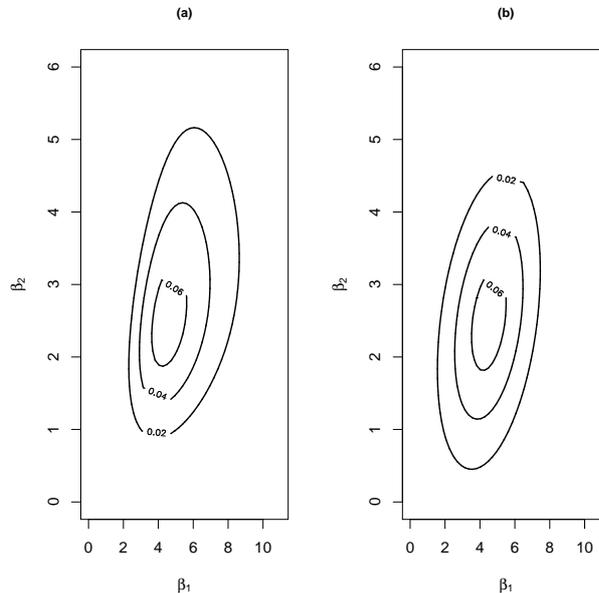


Figura 2: (a) Contorno de la distribución final de β en (55). (b) Contorno de la aproximación en (56) de la distribución final de β .

Se obtuvieron 3 cadenas de Markov a través de la implementación del MGRY, cada una con longitud 10,000. Los puntos de inicio $(10, 3)^t$, $(5, 8)^t$ y $(2, 1)^t$ se obtuvieron muestreando la distribución normal truncada $N_{\mathcal{T}}(\mathbf{0}, 100 \times \mathbf{I})$, iniciando con el punto:

$$\mathbf{B}^{-}\mathbf{b} = (12.34, 5.21)^t$$

donde \mathbf{B}^{-} es la inversa generalizada de la matriz \mathbf{B} , se generó una cadena de tamaño 10,000 eligiendo los 3 puntos con saltos igual a 3,333.

Se implementó el GDS, utilizando los mismos puntos de inicio para generar 3 cadenas de Markov, cada una con longitud 10,000.

En la Figura 3 se muestran las primeras 500 iteraciones de cada una de las cadenas para el GTMI y GDS. Los triángulos en las gráficas muestran los puntos de inicio de cada cadena. En esta Figura se observa que el GDS cubre un poco más la región \mathcal{T} que el GTMI en cada una de las cadenas, note que la correlación entre las componentes del vector β es 0.33.

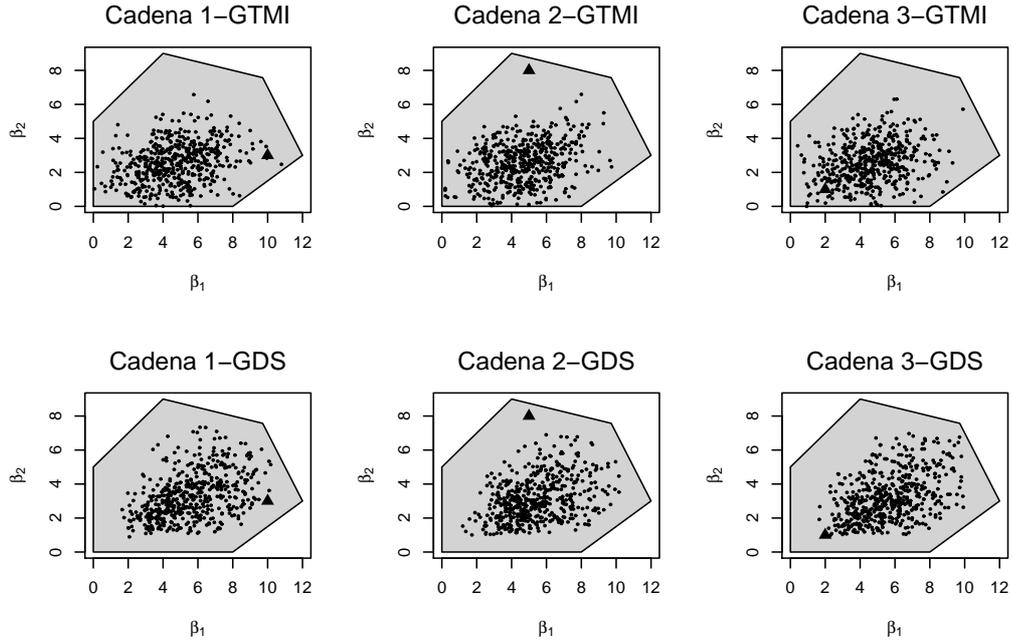


Figura 3: Primeras 500 iteraciones de cada cadena, generados por el GTMI y el GDS.

En el Cuadro 2 se muestra el diagnóstico de Raftery y Lewis para las componentes de β por cadena y método. El diagnóstico se obtuvo para $q = 0.025$, $r = 0.005$ y $s = 0.95$. En este cuadro se observa que para cada componente de β , el número de iteraciones que se requiere para estimar el cuantil $q = 0.025$ en cada una de las cadenas es más pequeño con el GTMI que con el GDS, lo mismo ocurre con el factor de dependencia I .

Cuadro 2: Diagnóstico de convergencia de Raftery y Lewis para β_1 y β_2 .

β	Cadena	Método	Burn-in (M)	Total (N)	Lower bound ($Nmin$)	Dependence factor (I)
β_1	1	GTMI	2	3680	3746	0.98
		GDS	2	3897	3746	1.04
	2	GTMI	2	3710	3746	0.99
		GDS	3	4095	3746	1.09
	3	GTMI	2	3620	3746	0.97
		GDS	3	4267	3746	1.14
β_2	1	GTMI	2	3710	3746	0.99
		GDS	2	3802	3746	1.01
	2	GTMI	2	3650	3746	0.97
		GDS	2	3929	3746	1.05
	3	GTMI	2	3680	3746	0.98
		GDS	2	3897	3746	1.04

En la Figura 4 se muestran las estimaciones \hat{R} para el factor de escala correspondiente a las componentes de β . Se utilizó el estimador del factor de escala dado en (9) para monitorearlo cada 20 iteraciones hasta llegar a la iteración 1,000. De acuerdo a esta figura, en ambos métodos las estimaciones del factor de escala alcanzan el valor de 1 un poco antes de la iteración 500, además, las estimaciones están por debajo de 1.01 casi desde el principio, lo cual hace evidente una convergencia rápida hacia las distribuciones $\pi(\beta|\mathbf{y})$ y $\pi_a(\beta|\mathbf{y})$. Se observa también que las estimaciones del factor de escala correspondientes al GTMI siempre están por debajo de las del GDS en ambos componentes de β .

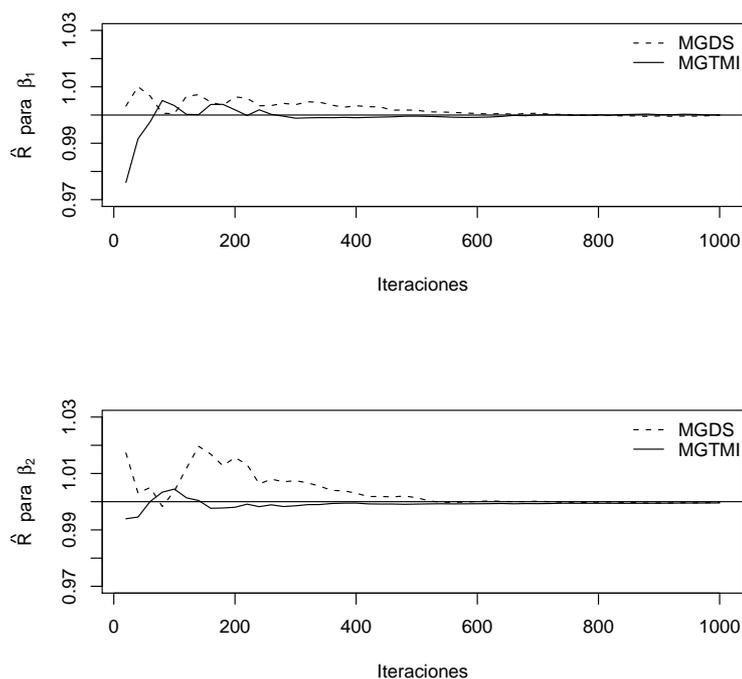


Figura 4: Factor de escala para β_1 y β_2 por método.

Las autocorrelaciones en la muestra de un Gibbs sampler pueden ser usadas para medir el desarrollo de una simulación. En Chen *et al.*, (2000), mencionan que el lento decaimiento en la función de autocorrelación sugiere una convergencia lenta dentro de una cadena y usualmente una convergencia lenta hacia la distribución de interés. En las Figuras 5 y 6 se muestra la función de autocorrelación para cada componente del vector β en la cadena 1, correspondiente a los métodos GTMI y GDS, respectivamente. En ambas gráficas se observa un decaimiento rápido en la función de autocorrelación correspondiente a cada componente del vector β , lo cual sugiere que ningún método tiene problema de convergencia hacia su respectiva distribución. No obstante, en ambas figuras se observa que el método GDS converge más lento que el GTMI.

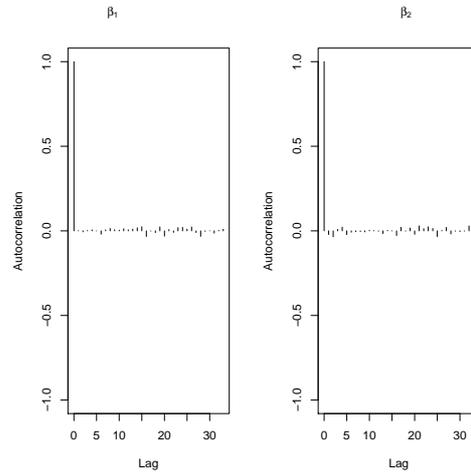


Figura 5: Autocorrelaciones para β_1 y β_2 en la cadena 1 del GTMI.

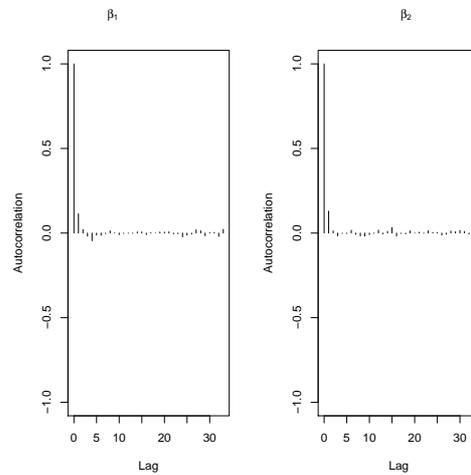


Figura 6: Autocorrelaciones para β_1 y β_2 en la cadena 1 del GDS.

En el Cuadro 3 se muestran las estimaciones de las componentes del vector β obtenida por cada método. En este cuadro se observa que la estimación correspondiente al verdadero valor de β tiene un sesgo promedio en valor absoluto de 1.01 con el GTMI y de 0.93 con el GDS.

Cuadro 3: Estimación, desviación estándar, valor verdadero del parámetro y sesgo para cada componente del vector β .

β	Método	Estimación	Desv. Est.	Sesgo
$\beta_1 = 4$	GTMI	5.79	2.00	-1.79
	GDS	5.64	1.82	-1.64
$\beta_2 = 3$	GTMI	3.22	1.48	-0.22
	GDS	3.21	1.30	-0.21

5.1.1.2. Resultados para $n = 100$

Para el caso $n = 100$, se simularon 100 datos con distribución Bernoulli con probabilidad $\mu_i = 1/(1 + \exp(-\mathbf{x}_i^t \boldsymbol{\beta}))$ donde \mathbf{X} es una matriz de 100×2 .

Se implementó el GTMI bajo las mismas condiciones que en el caso $n = 30$ y se obtuvo la media, la matriz de covarianzas y la matriz de correlaciones de $\pi_a(\boldsymbol{\beta}|\mathbf{y})$ dadas por

$$\hat{\boldsymbol{\beta}} = \begin{pmatrix} 3.7 \\ 2.46 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 0.68 & 0.39 \\ 0.39 & 0.39 \end{pmatrix}, \quad \boldsymbol{\rho} = \begin{pmatrix} 1 & 0.76 \\ 0.76 & 1 \end{pmatrix}$$

En la implementación de los métodos GTMI y GDS se consideraron los mismos puntos de inicio que en el caso $n = 30$, se obtuvieron 3 cadenas de Markov cada una con longitud 10,000.

En la Figura 7 se muestran las estimaciones \hat{R} para el factor de escala correspondiente a los componentes de $\boldsymbol{\beta}$. En esta figura se observa que el GTMI alcanza el valor de 1 desde la iteración 200 y converge mucho más rápido que el GDS, aunque las estimaciones \hat{R} están por debajo de 1.03 para ambos métodos desde el principio.

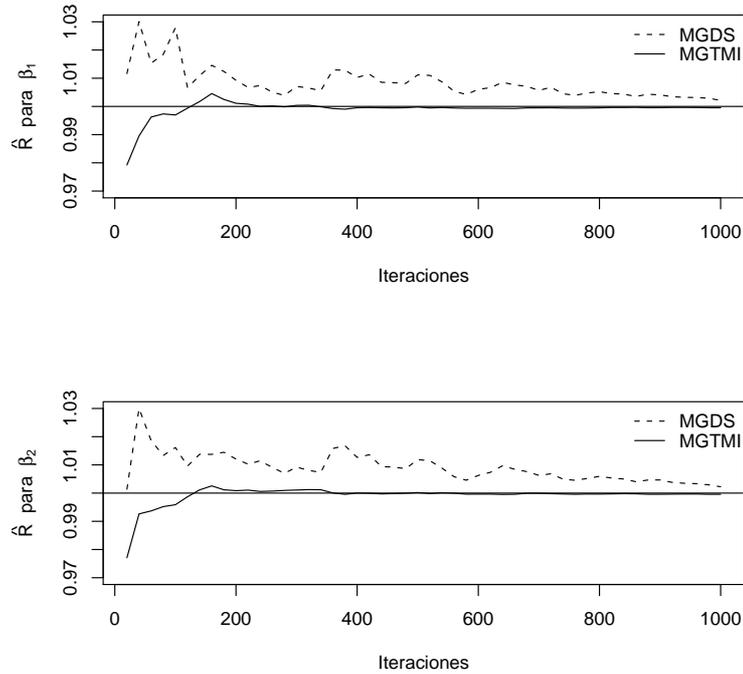


Figura 7: Factor de escala para β_1 y β_2 por método.

Las Figuras 8 y 9 muestran la función de autocorrelación para cada componente del vector $\boldsymbol{\beta}$ en la cadena 1, correspondiente a ambos métodos. En ambas figuras se observa que ningún método tiene problema de convergencia hacia su respectiva distribución. Sin embargo, en ambas figuras se observa que el método GDS converge más lento que el GTMI.

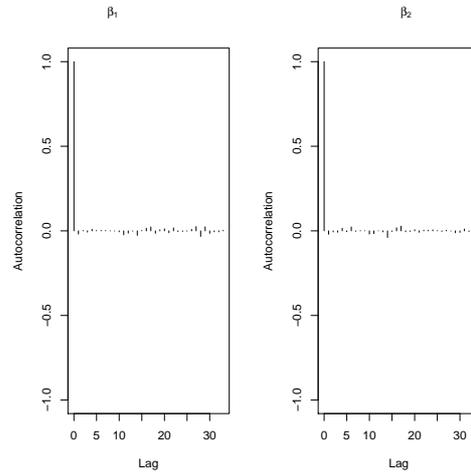


Figura 8: Autocorrelaciones para β_1 y β_2 en la cadena 1 del GTMI.

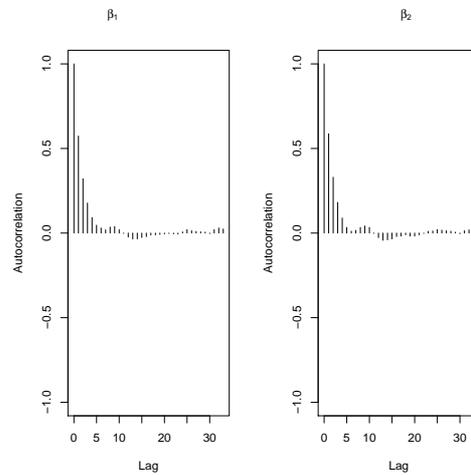


Figura 9: Autocorrelaciones para β_1 y β_2 en la cadena 1 del GDS.

En el Cuadro 4 se muestran las estimaciones de las componentes del vector β obtenida por cada método. En este cuadro se observa que la estimación correspondiente al verdadero valor de β es casi la misma en ambos métodos, obteniéndose un sesgo promedio en valor absoluto de 0.18 para ambos métodos.

Cuadro 4: Estimación, desviación estándar, valor verdadero del parámetro y sesgo para cada componente del vector β .

β	Método	Parámetro	Estimación	Desv. Est.	Sesgo
$\beta_1 = 4$	GTMI		4.02	0.90	-0.02
	GDS		4.01	0.88	-0.01
$\beta_2 = 3$	GTMI		2.66	0.66	0.34
	GDS		2.66	0.65	0.34

5.1.2. Caso 2: $\beta = (4.5, 8.3)^t$

5.1.2.1. Resultados para $n = 30$ y $n = 100$

Análogamente al caso 1 se implementaron los métodos GTMI y GDS, en la implementación del GTMI se utilizó el mismo punto de inicio para la moda que en el caso 1. En el Cuadro 5 se muestra el vector de medias $\hat{\beta}$, la matriz de covarianzas Σ y la matriz de correlaciones correspondiente a la distribución $\pi_a(\beta|\mathbf{y})$ para cada tamaño de muestra.

Cuadro 5: Vector de medias, matriz de covarianzas y matriz de correlaciones para $\pi_a(\beta|\mathbf{y})$.

n	$\hat{\beta}$	Σ		ρ	
30	$(5.05, 7.09)^t$	10.97	13.44	1.00	0.95
		13.44	18.10	0.95	1.00
100	$(7.06, 11.39)^t$	5.71	8.33	1.00	0.91
		8.33	14.69	0.91	1.00

En la Figura 10 se muestran las primeras 500 iteraciones de cada una de las cadenas para el GTMI y GDS, para el caso $n = 30$. De acuerdo a esta figura el GTMI cubre un poco más la región \mathcal{T} que el GDS en cada una de las cadenas, note que en este caso la correlación entre las componentes del vector β es 0.95 (Cuadro 5). Se obtiene el mismo resultado para el tamaño de muestra $n = 100$ (se omite la figura).

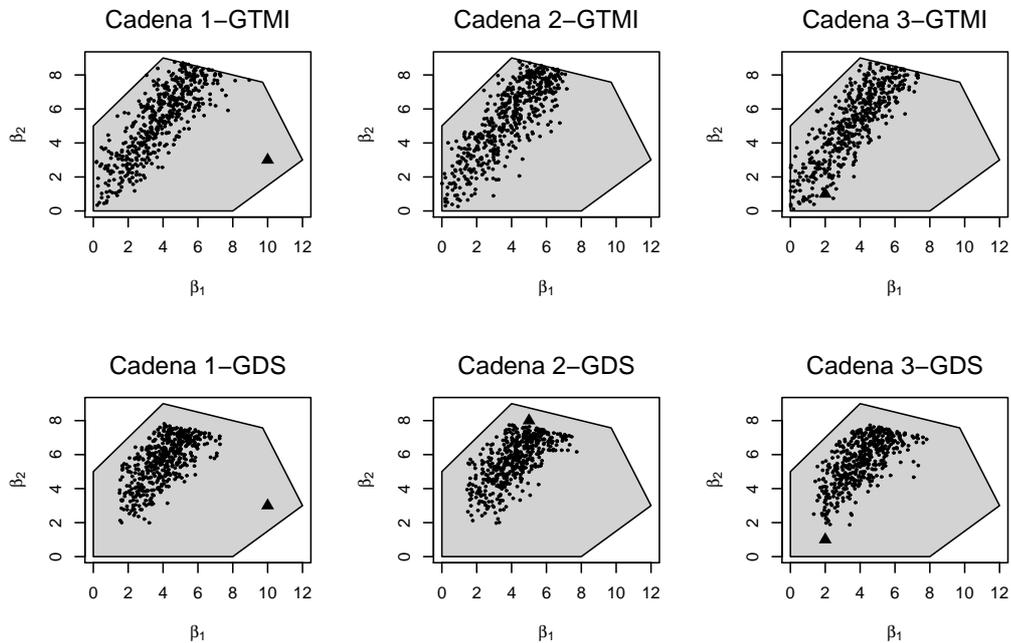


Figura 10: Primeras 500 iteraciones de cada cadena, generados por el GTMI y el GDS, $n = 30$.

Las Figuras 11 y 12 muestran la función de autocorrelación para cada componente del vector β en la cadena 1, correspondiente a ambos métodos para el tamaño de muestra $n = 30$. Se observa

que ningún método tiene problema de convergencia, no obstante el método GDS converge más lento que el GTMI. Para el tamaño de muestra 100 se da el mismo resultado (se omiten las figuras).

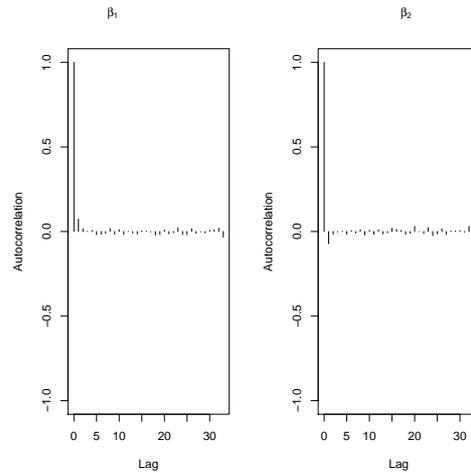


Figura 11: Autocorrelaciones para β_1 y β_2 en la cadena 1 del GTMI, $n = 30$.

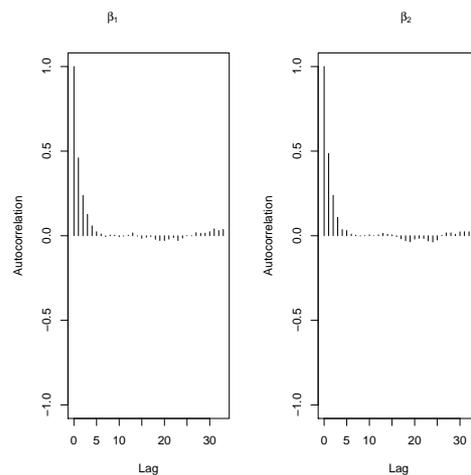


Figura 12: Autocorrelaciones para β_1 y β_2 en la cadena 1 del GDS, $n = 30$.

En el Cuadro 6 se muestran las estimaciones de las componentes del vector β obtenida por tamaño de muestra y método. En este cuadro se observa que la estimación obtenida con el GTMI para el verdadero valor de β tiene un sesgo promedio en valor absoluto de 1.24 y de 1.52 con el GDS para $n = 30$. Para $n = 100$ la estimación tiene un sesgo promedio en valor absoluto de 0.61 con el GTMI y de 0.83 con el GDS.

Cuadro 6: Estimación, desviación estándar, valor verdadero del parámetro y sesgo para cada componente del vector β , para $n = 30$ y $n = 100$.

n	β	Método	Estimación	Desv. Est.	Sesgo
30	$\beta_1 = 4.5$	GTMI	4.29	1.40	0.21
		GDS	4.12	1.25	0.38
	$\beta_2 = 8.3$	GTMI	6.04	1.59	2.26
		GDS	5.65	1.30	2.65
100	$\beta_1 = 4.5$	GTMI	4.79	0.92	-0.29
		GDS	4.49	0.92	0.01
	$\beta_2 = 8.3$	GTMI	7.38	0.98	0.92
		GDS	6.65	0.80	1.65

5.1.3. Caso 3: $\beta = (6.2, 0.8)^t$

5.1.3.1. Resultados para $n = 30$ y $n = 100$

En la implementación del GTMI se utilizó el mismo punto de inicio para la moda que en los casos 1 y 2. En el Cuadro 7 se muestra el vector de medias $\hat{\beta}$, la matriz de varianzas Σ y la matriz de correlaciones correspondiente a la distribución $\pi_a(\beta|\mathbf{y})$ para cada tamaño de muestra.

Cuadro 7: Vector de medias y matriz de varianzas y covarianzas para $\pi_a(\beta|\mathbf{y})$.

n	$\hat{\beta}$	Σ		ρ	
30	$(6.52, 0.66)^t$	10.97	1.19	1.00	0.53
		1.19	0.46	0.53	1.00
100	$(6.19, 0.72)^t$	2.28	0.33	1.00	0.52
		0.33	0.18	0.52	1.00

En la Figura 13 se muestran las primeras 500 iteraciones de cada una de las cadenas para el GTMI y GDS, para el caso $n = 30$. De acuerdo a esta figura el GTMI cubre un poco más la región \mathbf{T} que el GDS en cada una de las cadenas, note que en este caso la correlación entre las componentes del vector β es 0.53 (Cuadro 7). Se obtiene el mismo resultado para el tamaño de muestra $n = 100$ (se omite la figura).

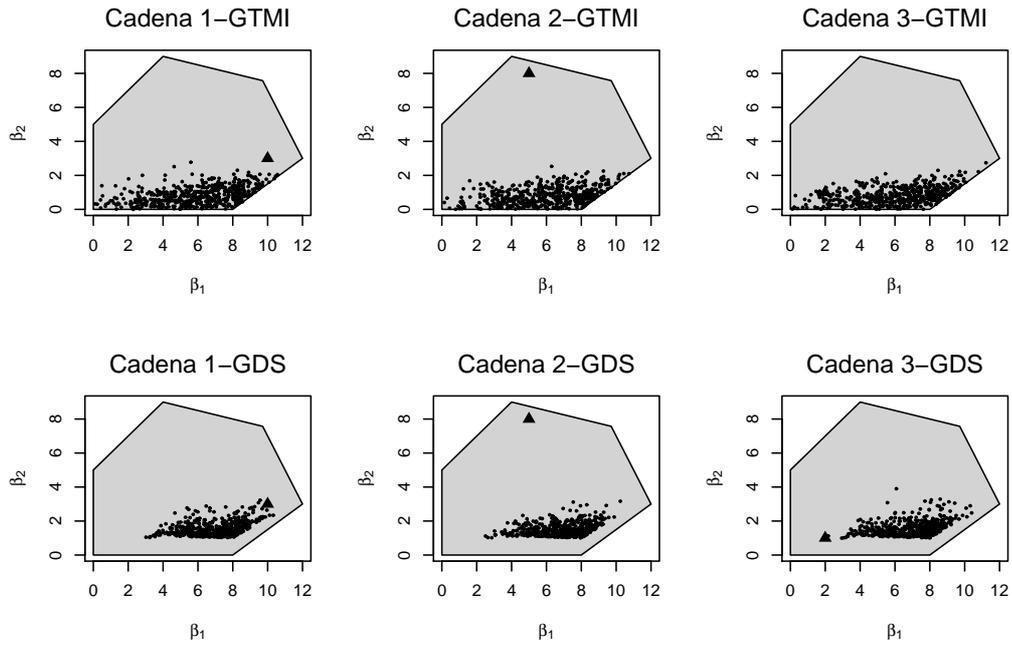


Figura 13: Primeras 500 iteraciones de cada cadena, generados por el GTMI y el GDS, $n = 30$.

Las Figuras 14 y 15 muestran la función de autocorrelación para cada componente del vector β en la cadena 1, correspondiente a ambos métodos para el tamaño de muestra $n = 30$. Se observa que ningún método tiene problema de convergencia, no obstante el método GDS converge más lento que el GTMI. Para el tamaño de muestra 100 se da el mismo resultado (se omiten las figuras).

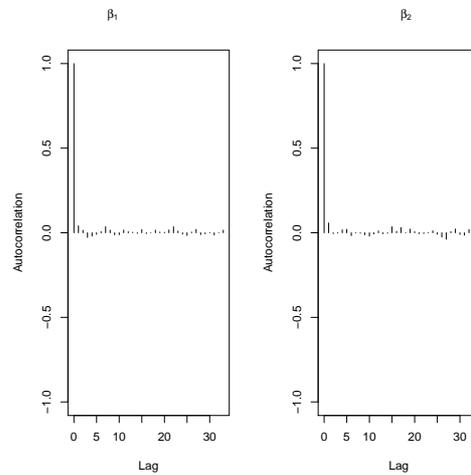


Figura 14: Autocorrelaciones para β_1 y β_2 en la cadena 1 del GTMI, $n = 30$.

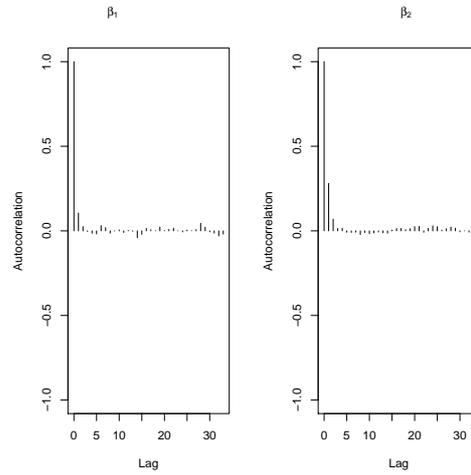


Figura 15: Autocorrelaciones para β_1 y β_2 en la cadena 1 del GDS, $n = 30$.

En el Cuadro 8 se muestran las estimaciones de las componentes del vector β obtenida por tamaño de muestra y método. En este cuadro se observa que la estimación obtenida con el GTMI para el verdadero valor de β tiene un sesgo promedio en valor absoluto de 0.29 y de 0.69 con el GDS para $n = 30$. Para $n = 100$ la estimación tiene un sesgo promedio en valor absoluto de 0.21 con el GTMI y de 0.71 con el GDS.

Cuadro 8: Estimación, desviación estándar, valor verdadero del parámetro y sesgo para cada componente del vector β , para $n = 30$ y $n = 100$.

n	β	Método	Estimación	Desv. Est.	Sesgo
30	$\beta_1 = 6.2$	GTMI	6.64	1.87	-0.44
		GDS	6.85	1.50	-0.65
	$\beta_2 = 0.8$	GTMI	0.94	0.59	-0.14
		GDS	1.53	0.40	-0.73
100	$\beta_1 = 6.2$	GTMI	6.62	1.35	-0.42
		GDS	7.11	0.97	-0.91
	$\beta_2 = 0.8$	GTMI	0.80	0.39	0.00
		GDS	1.30	0.21	-0.5

5.2. Distribución Exponencial

La función de distribución de las observaciones que se considera en esta Sección es la distribución exponencial, i.e.,

$$\begin{aligned} f(y_i) &= \frac{1}{\mu_i} \exp\left(-\frac{y_i}{\mu_i}\right), \quad y_i = 0, 1 \quad i = 1, \dots, n \\ &= \exp\left[-\frac{y_i}{\mu_i} - \log(\mu_i)\right], \end{aligned} \quad (57)$$

con $E(y_i) = \mu_i = 1/\lambda_i$. Se consideró la liga $g(\mu_i) = \log \mu_i$, con $g(\mu_i) = \eta_i = \mathbf{x}_i^t \boldsymbol{\beta}$.

Se consideró que el vector $\boldsymbol{\beta}$ sea de orden 5 y esté sujeto al conjunto de restricciones lineales:

$$\begin{aligned} \beta_1 \geq 0, \quad \beta_2 \geq 0, \quad \beta_3 \geq 0, \quad \beta_4 \geq 0, \quad \beta_5 \geq 0, \quad 2\beta_1 - 2\beta_2 + \beta_3 \leq 3, \\ -\beta_1 + \beta_2 - \beta_3 + \beta_4 - \beta_5 \leq 5, \quad \text{y} \quad -3\beta_1 + 2\beta_2 \leq 2. \end{aligned}$$

El conjunto \mathbf{T} en (25) para este conjunto de restricciones es:

$$\mathbf{T} := \{ \boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5) \in \mathfrak{R}^5 \mid \mathbf{B}\boldsymbol{\beta} \leq \mathbf{b} \} \quad (58)$$

donde

$$\mathbf{B} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 2 & -2 & 1 & 0 & 0 \\ -1 & 1 & -1 & 1 & -1 \\ -3 & 2 & 0 & 0 & 0 \end{bmatrix} \quad \text{y} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 3 \\ 5 \\ 2 \end{bmatrix}.$$

Se consideran los siguientes valores para el vector $\boldsymbol{\beta}$ en la región (58):

$$(3.09, 4.05, 4.30, 2.52, 0.85)^t \quad \text{y} \quad (2.32, 2.74, 1.01, 2.57, 0.79)^t$$

5.2.1. Caso 1: $\boldsymbol{\beta} = (3.09, 4.05, 4.30, 2.52, 0.85)^t$

5.2.1.1. Resultados para $n = 30$

Para el caso $n = 30$, se simularon 30 datos exponenciales con parámetros de forma $\lambda_i = 1/\mu_i = 1/e^{\eta_i}$, donde \mathbf{X} es una matriz de orden 30×5 .

En el Cuadro 9 se muestran los 30 datos junto con los valores de las cinco covariables que se usaron en este modelo.

Cuadro 9: Covariables y realizaciones del modelo en (57).

Num.	\mathbf{y}	\mathbf{x}_1	\mathbf{x}_2	\mathbf{x}_3	\mathbf{x}_4	\mathbf{x}_5
1	5559.118	1.466	0.288	-0.228	0.682	1.372
2	0.004	-0.951	-1.241	-0.222	1.526	1.469
3	0.001	0.295	-1.612	0.147	-0.213	-0.090
4	7312.960	-0.584	0.234	1.671	1.060	0.341
5	3.139	0.750	-1.478	1.064	1.21	-0.105
6	242922.7	0.990	-0.631	2.257	0.771	2.506
7	3.893	1.241	-0.687	-0.510	0.951	0.917
8	0.004	-1.548	-0.112	0.565	-0.195	-0.112
9	2037.74	0.758	1.016	0.519	-1.109	2.198
10	12.046	0.096	0.848	1.008	-0.409	-0.755
11	32.31	-0.415	1.031	0.801	-0.508	-0.588
12	6025672	-0.234	1.740	1.574	0.670	0.409
13	0.707	0.235	1.098	-1.180	-0.746	1.101
14	0.002	-1.033	0.153	-0.224	-0.858	-0.008
15	0.176	0.319	-1.573	1.141	-0.095	-0.169
16	5.751	0.940	0.463	-1.107	1.157	-1.083
17	0.002	-0.466	0.375	-0.971	-0.125	1.180
18	4266.041	0.629	1.237	0.383	0.166	1.702
19	23.008	0.953	0.266	0.460	-0.169	0.614
20	307.465	0.464	0.437	0.210	0.751	1.732
21	481.090	0.064	-0.271	1.992	-0.410	-0.966
22	0.004	0.808	-0.171	-1.205	-0.095	1.274
23	0.000	-1.331	-0.864	-2.957	1.275	0.480
24	160.395	0.037	1.037	-0.024	0.675	0.386
25	2.657	-1.789	1.591	0.385	-1.106	0.553
26	5.123	-0.220	-0.507	1.403	0.533	-1.010
27	0.112	0.295	-0.608	-0.340	-0.243	0.756
28	1.048	-0.779	0.776	1.001	-1.182	-0.514
29	83.893	-0.261	0.426	1.400	-0.103	-2.459
30	4.707	0.615	0.405	-0.224	-0.136	0.920

La función de verosimilitud está dada por

$$L(\boldsymbol{\beta}; \mathbf{y}) \propto \exp \left[- \sum_{i=1}^{30} y_i e^{-\mathbf{x}_i^t \boldsymbol{\beta}} - \sum_{i=1}^{30} \mathbf{x}_i^t \boldsymbol{\beta} \right] I_{\mathcal{T}}(\boldsymbol{\beta})$$

y las componentes w_i y σ_i^2 en (30) y (31) son

$$\begin{aligned} w_i &= \hat{\eta}_i + \frac{y_i e^{-\eta_i} - 1}{y_i e^{-\eta_i}} \\ \sigma_i^2 &= \frac{1}{y_i e^{-\eta_i}} \end{aligned}$$

La distribución inicial para el vector $\boldsymbol{\beta}$ es la no informativa, i.e., $\pi(\boldsymbol{\beta}) = \mathbf{1}_{\mathbf{T}}(\boldsymbol{\beta})$, de donde la distribución final del vector $\boldsymbol{\beta}$ es

$$\pi(\boldsymbol{\beta} | \mathbf{y}) \propto L(\boldsymbol{\beta}; \mathbf{y}) \propto \exp \left[- \sum_{i=1}^{30} y_i e^{-\mathbf{x}_i^t \boldsymbol{\beta}} - \sum_{i=1}^{30} \mathbf{x}_i^t \boldsymbol{\beta} \right] I_{\mathbf{T}}(\boldsymbol{\beta}) \quad (59)$$

Para la implementación del GTMI se eligió un punto al azar en la región (58) como valor de inicio para la obtención de la moda de la distribución en (59), obteniéndose el punto $(8.89, 12.98, 10.33, 3.95, 12.94)^t$ y con ello, la aproximación normal $\pi_a(\boldsymbol{\beta} | \mathbf{y})$ para el vector $\boldsymbol{\beta}$ tiene vector de medias $\hat{\boldsymbol{\beta}} = (3.36, 4.41, 4.20, 2.23, 0.580)^t$ y matriz de covarianzas

$$\boldsymbol{\Sigma} = \begin{pmatrix} 0.06 & 0.02 & -0.00 & -0.01 & -0.02 \\ 0.02 & 0.06 & -0.01 & 0.02 & -0.02 \\ -0.00 & -0.01 & 0.03 & -0.01 & 0.01 \\ -0.01 & 0.02 & -0.01 & 0.08 & -0.01 \\ -0.02 & -0.02 & 0.01 & -0.01 & 0.05 \end{pmatrix}$$

Para muestrear la distribución $\pi_a(\boldsymbol{\beta} | \mathbf{y})$ se eligieron los puntos de inicio $(0.88, 0.58, 0.42, 0.07, 1.64)$, $(1.42, 0.56, 1.11, 1.28, 1.57)$ y $(2.08, 1.98, 0.34, 0.18, 3.07)$ en la región \mathbf{T} de (58), con los cuales se obtuvieron 3 cadenas de Markov a través de la implementación del MGRY, cada una con longitud 10,000. Los puntos de inicio se obtuvieron muestreando la distribución normal truncada $N_{\mathbf{T}}(\mathbf{0}, 100 \times \mathbf{I})$, iniciando con el punto:

$$\mathbf{B}^{-1} \mathbf{b} = (-0.65, -0.60, 0.62, 1.86, -1.86)^t.$$

Se generó una cadena de tamaño 10,000 eligiendo los 3 puntos con saltos igual a 3,333.

En el Cuadro 10 se muestra el diagnóstico de Raftery y Lewis para cada una de las componentes del vector $\boldsymbol{\beta}$. El diagnóstico se obtuvo para $q = 0.025$, $r = 0.005$ y $s = 0.95$. En general se tiene un factor de dependencia muy cercano a 1 y un número promedio de 3,764 iteraciones por cadena para estimar el cuantil $q = 0.025$.

Cuadro 10: Diagnóstico de Raftery y Lewis para las componentes de β .

Cadena	β	Burn-in (M)	Total (N)	Lower bound ($Nmin$)	Dependence factor (I)
1	β_1	2	3821	3746	1.02
	β_2	2	3788	3746	1.01
	β_3	2	3741	3746	1.00
	β_4	2	3771	3746	1.01
	β_5	2	3650	3746	0.97
2	β_1	2	3882	3746	1.04
	β_2	2	3947	3746	1.05
	β_3	2	3771	3746	1.01
	β_4	2	3680	3746	0.98
	β_5	2	3620	3746	0.97
3	β_1	2	3741	3746	1.00
	β_2	2	3851	3746	1.03
	β_3	2	3834	3746	1.02
	β_4	2	3650	3746	0.97
	β_5	2	3710	3746	0.99

En la Figura 16 se muestran las estimaciones \hat{R} del factor de escala para las componentes de β . El factor de escala se monitorea cada 20 iteraciones hasta llegar a la iteración 2,000. De acuerdo a esta figura, las estimaciones del factor de escala se encuentran a dos centésimas de 1 a partir de la iteración 250, lo cual hace evidente una convergencia rápida hacia la distribución $\pi_a(\beta|\mathbf{y})$.

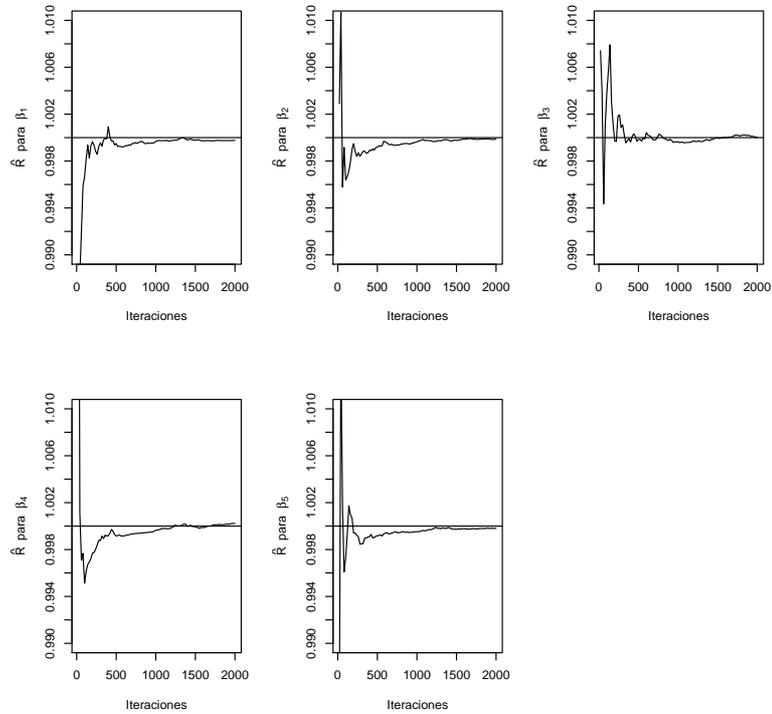


Figura 16: Factor de escala para las componentes de β .

En la Figura 17 se muestra la función de autocorrelación para cada componente del vector β obtenido por el GTMI. En esta figura se observa un decaimiento rápido en las autocorrelaciones correspondiente a cada componente del vector β .

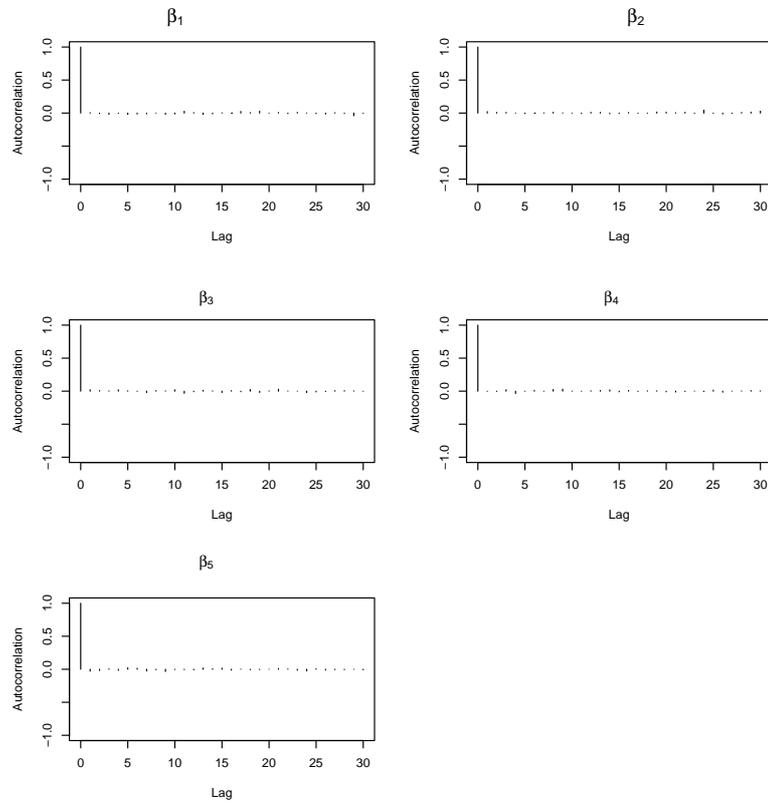


Figura 17: Autocorrelaciones para las componentes de β .

En el Cuadro 11 se muestran las estimaciones de las componentes del vector β obtenida por cada método. En este cuadro se observa que la estimación correspondiente al verdadero valor de β tiene un sesgo promedio en valor absoluto de 0.26.

Cuadro 11: Estimación, desviación estándar, valor verdadero del parámetro y sesgo para cada componente del vector β .

β	Estimación	Desv. Est.	Sesgo
$\beta_1 = 3.09$	3.30	0.25	-0.21
$\beta_2 = 4.05$	4.43	0.23	-0.38
$\beta_3 = 4.30$	4.04	0.17	0.26
$\beta_4 = 2.52$	2.28	0.29	0.24
$\beta_5 = 0.85$	0.63	0.21	0.22

5.2.1.2. Resultados para $n = 100$ y $n = 400$

Para la implementación del GTMI se utilizó el mismo punto de inicio para la moda que en el caso $n = 30$. En el Cuadro 12 se muestra el vector de medias $\hat{\beta}$ y la matriz Σ para la distribución resultante $\pi_a(\beta|\mathbf{y})$ para cada tamaño de muestra.

Cuadro 12: Vector de medias y matriz de varianzas y covarianzas para $\pi_a(\beta|\mathbf{y})$.

n	$\hat{\beta}$	Σ				
100	$(3.06, 4.01, 4.28, 2.44, 0.86)^t$	1.12e-02	-0.22e-02	0.15e-03	-0.74e-03	4.87e-05
		-2.27e-03	1.01e-02	0.55e-03	0.16e-02	-2.07e-03
		1.48e-04	0.55e-03	1.12e-02	-0.501e-03	-3.50e-03
		-0.741e-03	0.16e-02	-0.501e-03	1.08e-02	2.74e-03
		4.87e-05	-0.21e-02	-0.35e-02	0.27e-02	1.58e-02
400	$(3.16, 4.00, 4.30, 2.49, 0.85)^t$	2.39e-03	-2.02e-04	-6.79e-05	-6.094e-05	3.67e-05
		-2.02e-04	2.63e-03	1.60e-04	5.96e-05	-1.11e-04
		-6.79e-05	1.60e-04	2.62e-03	3.57e-05	2.71e-05
		-6.09e-05	5.96e-05	3.57e-05	2.48e-03	-9.02e-05
		3.67e-05	-1.11e-04	2.71e-05	-9.02e-03	2.38e-03

En el Cuadro 13 se muestran las estimaciones de las componentes del vector β obtenida por cada método. En este cuadro se observa que la estimación correspondiente al verdadero valor de β tiene un sesgo promedio en valor absoluto de 0.03 para $n = 100$ y $n = 400$.

Cuadro 13: Estimación, desviación estándar y sesgo para cada componente del vector β .

n	β	Estimación	Desv. Est.	Sesgo
100	$\beta_1 = 3.09$	3.06	0.10	0.03
	$\beta_2 = 4.05$	4.02	0.10	0.03
	$\beta_3 = 4.30$	4.29	0.11	0.01
	$\beta_4 = 2.52$	2.45	0.10	0.07
	$\beta_5 = 0.85$	0.87	0.12	-0.02
400	$\beta_1 = 3.09$	3.16	0.05	-0.07
	$\beta_2 = 4.05$	4.00	0.05	0.05
	$\beta_3 = 4.30$	4.30	0.05	0.00
	$\beta_4 = 2.52$	2.49	0.05	0.03
	$\beta_5 = 0.85$	0.85	0.05	0.00

5.2.2. Caso 2: $\beta = (2.32, 2.74, 1.01, 2.57, 0.79)^t$

5.2.2.1. Resultados para $n = 30$, $n = 100$ y $n = 400$

Para la implementación del GTMI se utilizó el mismo punto de inicio para la moda que en el caso 1. En el Cuadro 14 se muestra el vector de medias $\hat{\beta}$ y la matriz Σ para la distribución resultante $\pi_a(\beta|\mathbf{y})$ para cada tamaño de muestra.

Cuadro 14: Vector de medias y matriz covarianzas para $\pi_a(\boldsymbol{\beta}|\mathbf{y})$.

n	$\hat{\boldsymbol{\beta}}$	$\boldsymbol{\Sigma}$				
30	(2.18, 2.63, 1.23, 2.69, 0.67) ^t	3.29e-02	-1.21e-05	-2.56e-03	0.11e-02	0.29e-02
		-1.21e-05	3.74e-02	-3.89e-03	-0.65e-02	-1.30e-02
		-2.56e-03	-3.89e-03	4.57e-02	-1.35e-02	0.60e-02
		1.14e-03	-6.52e-03	-1.35e-02	2.73e-02	-0.38e-03
		2.90e-03	-1.30e-02	0.60e-02	-0.38e-03	4.95e-02
100	(2.43, 2.78, 0.94, 2.48, 0.79) ^t	1.43e-02	0.24e-03	0.16e-02	-0.38e-02	-0.64e-03
		0.24e-03	0.97e-02	0.10e-02	0.89e-03	0.12e-02
		0.16e-02	0.10e-02	0.93e-02	0.13e-02	0.43e-03
		-0.38e-02	0.890e-03	0.13e-02	1.86e-02	-0.21e-02
		-0.64e-03	0.12e-02	0.43e-03	-0.21e-02	1.45e-02
400	(2.34, 2.75, 1.00, 2.62, 0.82) ^t	2.90e-03	-8.56e-05	-1.20e-04	6.88e-05	-6.40e-05
		-8.56e-05	2.51e-03	-8.75e-05	1.56e-04	-1.09e-04
		-1.20e-04	-8.75e-05	2.13e-03	-2.11e-04	7.624e-05
		6.88e-05	1.56e-04	-2.11e-04	2.64e-03	2.02e-05
		-6.40e-05	-1.09e-04	7.62e-05	2.02e-05	2.57e-03

En el Cuadro 15 se muestran las estimaciones de las componentes del vector $\boldsymbol{\beta}$ obtenida por cada método. En este cuadro se observa que la estimación correspondiente al verdadero valor de $\boldsymbol{\beta}$ tiene un sesgo promedio en valor absoluto de 0.13, 0.06 y 0.02 para $n = 30$, $n = 100$ $n = 400$, respectivamente.

Cuadro 15: Media, desviación estándar y sesgo para cada componente del vector $\boldsymbol{\beta}$.

n	$\boldsymbol{\beta}$	Media	Desv. Est.	Sesgo
30	$\beta_1 = 2.32$	2.19	0.18	0.13
	$\beta_2 = 2.74$	2.65	0.18	0.09
	$\beta_3 = 1.01$	1.23	0.21	-0.22
	$\beta_4 = 2.57$	2.67	0.16	-0.10
	$\beta_5 = 0.79$	0.66	0.21	0.13
100	$\beta_1 = 2.32$	2.43	0.12	-0.11
	$\beta_2 = 2.74$	2.79	0.10	-0.05
	$\beta_3 = 1.01$	0.94	0.10	0.07
	$\beta_4 = 2.57$	2.48	0.13	0.09
	$\beta_5 = 0.79$	0.79	0.12	0.00
400	$\beta_1 = 2.32$	2.34	0.05	-0.02
	$\beta_2 = 2.74$	2.75	0.05	-0.01
	$\beta_3 = 1.01$	1.00	0.05	0.01
	$\beta_4 = 2.57$	2.62	0.05	-0.05
	$\beta_5 = 0.79$	0.82	0.05	-0.03

6. CONCLUSIONES

En el primer estudio de simulación se planteó un modelo MLG Bernoulli con la función ligandica canónica logarítmica y con dos componentes para el vector de coeficientes de regresión. Se consideraron tres casos para el verdadero valor de β en un soporte restringido y para cada caso se consideraron los tamaños de muestra 30 y 100. En todos los casos se implementaron los métodos GTMI y GDS.

En la Figura 2 se observa que para el caso 1 con $n = 30$, la distribución $\pi_a(\beta|\mathbf{y})$ se aproxima bastante bien a la verdadera distribución de β , en general se obtuvo el mismo resultado en todos los casos (las figuras se omitieron).

El GTMI cubre más la región \mathbf{T} en las tres cadenas de Markov para los casos dos y tres (Figura 10 y 13), dando como resultado una mayor representatividad en la muestra Gibbs obtenida.

Basándose en los resultados mostrados en el Cuadro 2 se nota que para ambos componentes del vector, el factor de dependencia I obtenido por el GTMI es mas pequeño que el obtenido por el GDS. Además, el número de iteraciones que se requiere por cadena para estimar el cuantil $q = 0.025$ en promedio es menor para el GTMI que para el GDS. Este resultado se observó en todos los casos (las cuadros se omitieron).

En las Figuras 4 y 7 se nota que el GDS converge mucho más lento que el GTMI, esto también se puede corroborar con las Figuras correspondientes a las autocorrelaciones (Figuras 5, 6, 8, 9, 11, 12, 14 y 15), además, con el GTMI se obtienen muestras “casi” independientes.

Si se asume como función de pérdida el error cuadrático, entonces el estimador de Bayes para el vector de coeficientes de regresión β , es la media muestral. En las dos modalidades de tamaño de muestra para los casos 2 y 3, se obtuvo un sesgo promedio más pequeño para la media de β con el GTMI (Cuadros 6 y 8). En el caso 1 para $n = 30$, el sesgo promedio para la media de β es ligeramente menor con el GDS (0.93 vs. 1.01, Cuadro 3) y para $n = 100$ el sesgo promedio es igual en ambos métodos (0.18, Cuadro 4).

En este estudio se pudo notar que la eficiencia del GTMI resultó ser muy significativa con respecto a la del GDS, sobre todo para los tamaños de muestras mayores a 30; corroborando el hecho de que la implementación directa del Gibbs sampler en un problema de restricción no es computacionalmente muy eficiente.

En el segundo estudio de simulación, se planteó un modelo exponencial con la función ligandica logarítmica y con cinco componentes para el vector de coeficientes de regresión, en este caso la verosimilitud no es log-cóncava por lo que sólo se implementó el GTMI. Se consideraron dos casos para el verdadero valor de β . En general, se observó que el GTMI es de convergencia rápida

(Cuadro 10 y Figura 16) y se obtuvieron muestras “casi” independientes (Figura 17). En los dos casos y en sus tres modalidades de tamaño de muestra se obtuvo un sesgo pequeño para la media de β (Cuadros 11, 13, 15). El promedio para la longitud mínima por cadena en ambos casos fue de 3,764.

En general, en ambos estudios de simulación se pudo notar un buen desempeño del GTMI siendo lo más destacado su eficiencia.

Se realizaron otras pruebas que no se incluyeron en este trabajo y en éstas se pudo notar que si el tamaño de muestra es pequeño pero se tiene buena información para la distribución inicial del vector de coeficientes de regresión, el GTMI proporciona buenos resultados.

7. DISCUSIÓN

En este trabajo se propuso el GTMI, el cual es simple y no requiere de técnicas numéricas sofisticadas, este método permite el análisis Bayesiano para el vector de coeficientes de regresión en un MLG cuando éste se encuentra sujeto a un conjunto de restricciones lineales de desigualdades; la idea principal del método consiste en aproximar la verosimilitud del MLG por la verosimilitud normal de un modelo lineal, bajo estas condiciones se implementa directamente el MGRY para facilitar el muestreo de la distribución normal multivariada truncada la cual es una aproximación de la distribución aposteriori del vector de coeficientes de regresión, sugiriendo la técnica importancia de muestreo para obtener las inferencias Bayesianas de interés.

La importancia del GTMI consiste en que es fácil de implementar y se puede utilizar bajo condiciones de independencia, dependencia y cuando el número de restricciones lineales excede el número de parámetros de regresión, además, es de convergencia rápida y se reduce a muestrear condicionales univariadas normales truncadas. El GTMI es una alternativa razonable a un problema en el que la mayoría de las veces el análisis Bayesiano es sumamente complicado y la implementación directa del Gibbs sampler o del Metropolis-Hastings no es eficiente. Cabe mencionar, que el GTMI es una alternativa cuando el tamaño de muestra es grande.

Un uso potencial del GTMI es que puede servir de base para analizar el caso cuando en un MLG con restricciones en el vector de coeficientes de regresión se encuentra un parámetro de escala que es desconocido pero es de interés para el usuario, en este caso se aproxima la verosimilitud del MLG alrededor de la moda aposteriori, con una distribución normal en el parámetro de regresión condicionada en el parámetro de escala. Para la obtención de la moda del modelo Bayesiano marginal correspondiente al parámetro de escala se puede utilizar el algoritmo de Newton-Raphson o el algoritmo EM (por sus siglas en inglés, Expectation Maximization) y para la obtención de la moda del modelo Bayesiano para el parámetro de regresión condicionada en el de escala, se utiliza la metodología de regresión lineal ponderada. Para muestrear el modelo univariado correspondiente al parámetro de escala se puede implementar el algoritmo ARMS (por sus siglas en inglés, Adaptive Rejection Metropolis Sampling), mientras que para muestrear eficientemente el modelo Bayesiano aproximado correspondiente al parámetro de regresión se utiliza el GTMI; bajo estas condiciones se utiliza la misma metodología expuesta en este trabajo para la obtención de inferencias Bayesianas para el vector de coeficientes de regresión.

8. REFERENCIAS

1. Apostol, T. M. (1969). Calculus volume II multivariable calculus and linear algebra, with applications to differential equations and probability. Second edition. Ed. John Wiley & Sons. pp. 311-321.
2. Box, G. E. P. y Tiao, G. C. (1973). Bayesian inference in statistical analysis. Ed. John Wiley. p. 418.
3. Chaudhuri, S. y Perlman, M.D. (2003). On the bias and mean-square error of order-restricted maximum likelihood estimators. Technical report No. 437 Department of Statistical, University of Washington, Seattle.
4. Chen, M. H., Shao, Q. M. y Ibrahim, J. G. (2000). Monte Carlo methods in bayesian computation. Springer-Verlag New York, UAS. pp. 191-208.
5. Dellaportas, P. y Smith, A.F.M. (1993). Bayesian inference for generalized linear and proportional Hazards models via Gibbs sampling. *Appl. Statistics*. 42: 443-459.
6. Dopson, A. J. (1945). An introduction to generalized linear models. Second edition. Ed. Chapman & Hall/CRC. pp.
7. Devroye, L. (1986). Non-uniform random variate generation. New York, Springer. pp. 38, 287.
8. Dunson, D.B. y Neelon, B. (2002). Bayesian inference on order constrained parameters in generalized linear models. Biostatistics Branch, National Institute of Enviromental Health Sciences, MD A3-03.
9. Fosdick, L.D. (1963). Monte Carlo calculations on the ising lattice. *Meth. Comput. Phys.*, 1, 245-280.
10. Gelfand, A.E. y Smith, A.F.M. (1990). Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*. 85: 398-409.
11. Gelfand, A.E., Smith, A.F.M. y Lee, T.M. (1992). Bayesian analysis of constrained parameter and truncated data problems using Gibbs sampling. *Journal of the American Statistical Association*. 87: 523-532.

12. Gelman A. y Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*. 7: 457-511.
13. Gelman A., Carlin, J., Stern, H. y Rubin, D. (2004). *Bayesian data analysis*. 2nd Edn. Chapman and Hall/CRC.USA. pp. 294-297, 357.
14. Geman, S. y Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*. 6: 721-741.
15. Geweke, J. (1996). *Modeling and prediction: Honouring Seymour Geisser*, eds. W. O. Johnson, J. C. Lee, and A. Zellner, New York, Springer, pp. 248-263.
16. Geyer, C. J. (1991). Constrained maximum likelihood exemplified by isotonic convex logistic regression. *Journal of the American Statistical Association*, Vol. 86, No. 415, Theory and Methods.
17. Geyer, C. J. (1994). Estimating normalizing constants and reweighting mixtures in Markov chain Monte Carlo. Revision of technical report No. 568. School of statistics, University of Minnesota.
18. Gilks, W.R., Richardson, S. y Spiegelhalter, D.J. (1996). *Markov chain Monte Carlo in Practice*, *Interdisciplinary statistics*. London: Chapman & Hall. USA. pp. 5-7, 46.
19. Gilks, W.R. y Roberts, G.O. (1996). *Markov chain Monte Carlo in practice*(eds W.R. Gilks, S. Richardson and D.J. Spiegelhalter), *interdisciplinary statistics*. London: Chapman & Hall. pp. 89-114.
20. Gilks, W.R. y Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Appl. Statistics.*, 41, 337-348.
21. Givens, G. H. y Hoeting, J. A. (2005). *Computational statistics*, Wiley-Interscience. pp. 163-164.
22. Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*. 57: 97-109.
23. Kalbfleisch, J. G. (1985). *Probability and statistical inference*. Volume 2: *Statistical inference*. Second edition. Springer-Verlang, New York, New York, U.S.A. pp. 40-42.

24. Lewis, S. M. (1995). An S code translation of the Gibbsit FORTRAN program by Steven Lewis. This S function was developed by Steven Lewis from an earlier function originally written by Karen Vines. Developer: Steven M. Lewis (slewis@stat.washington.edu).
25. Liew, C. K. y Shim, J. K. (1976). Inequality constrained least-squares estimation. *Journal of the American Statistical Association*, 71, 746-751.
26. McCullagh, P. y Nelder, J.A. (1989). *Generalized linear models*. 2nd Edn. London: Chapman and Hall. New York, USA. p. 32.
27. McDonald, J. W. y Diamond, I. D. (1990). On the fitting of generalized linear models with nonnegativity parameter constraints. *Biometrics* 46, 201-206.
28. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. y Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of chemical physics*. 21: 1087-1091.
29. Mukerjee, H. y Tu, R. (1995). Order-restricted inferences in linear regression. *Journal of the American statistical association*, Vol. 90, No. 430, Theory and methods.
30. Nelder, J.A. y Wedderburn, R.W.M. (1972). *Generalized linear models*. *Journal of the Royal Statistical Society, Series A*, 135, 370-384.
31. R Development Core Team (2007). *R: A language and environment for statistical computing*. R Foundation for statistical computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
32. Raftery, A.E. y Lewis, S.M. (1992). One long run with diagnostics. Implementation strategies for Markov chain Monte Carlo. *Statistical science*, 7, 493-497.
33. Rockafellar, R.T. (1970). *Convex analysis*. Princeton: Princeton university press. p. 77
34. Rodríguez-Yam, G. (2003). P.H. D. Dissertation, Colorado State University, Ft. Collins, CO. Estimation for state-space models and bayesian regression analysis with parameter constraints.
35. Wedderburn, R. W. M. (1976). On the existence and uniqueness of the maximum likelihood estimates for certain generalized linear models. *Biometrika*, 63, 27-32.

36. Zeger, S. L. y Rezaul, M. K. (1991). Generalized linear models with random effects, a Gibbs sampling approach. *Journal of the american statistical association*. Vol. 86, No. 413, theory and methods.

ANEXO A. PROGRAMAS EN R

A1. Simulación de datos

En este anexo se muestra el programa escrito en lenguaje R (R Development Core Team, 2007), el cual se utilizó para simular los datos con distribución Bernoulli y exponencial.

```
#Simulación de datos Bernoulli
#Caso 1. Suponiendo que el valor verdadero valor del parámetro es (4,3)
n <- 30 #tamaño de la muestra
p <- 2 #Número de betas
betas <- array(c(4,3),c(p,1)) #Valor verdadero en la región de truncamiento
#simulación de la respuesta del modelo verdadero
X <- array(rnorm(n*p, mean=0, sd=1),c(n,p)) #simulación de la matriz de datos
qr(X)$rank #Calcula el rango de la matriz
nn <- array(rep(1,n), c(n,1)) #vector de n's
py <- array(rep(0,n),c(n,1))
for (i in 1:n){py[i] <- 1/(1+exp(-(X[i,]*betas)))}#vector de probabilidades
y <- rbinom(n,nn,py)

#Guardando los datos simulados
save(X,y,file="C:/Simulados/Bernoulli/Datos1Ber30.Rdata")
#####

#Simulación de datos Exponencial
#Caso 1. El valor verdadero es (3.0943914,4.0469090,4.3017821,2.5178921,0.8460264)
n <- 30 #tamaño de la muestra
p <- 5 #Número de betas
betas <- array(c(3.0943914,4.0469090,4.3017821,2.5178921,0.8460264),c(p,1))
#simulación de la respuesta del modelo verdadero
X <- array(rnorm(n*p, mean=0, sd=1),c(n,p)) #simulación de la matriz de datos
qr(X)$rank #Calcula el rango de la matriz
lambda <- array(rep(0,n),c(n,1))
for (i in 1:n){lambda[i] <- 1/exp(X[i,]*betas)}#vector de probabilidades
y <- rexp(n,lambda)

#Guardando los datos simulados
save(X,y,file="C:/Simulados/Exponencial/Datos1Exp30.Rdata")
```

A2. Elección de puntos de inicio

En este anexo se muestra el programa escrito en lenguaje R (R Development Core Team, 2007), el cual se utilizó para obtener puntos en el interior de la región de truncamiento que sirven de arranque para las cadenas. Se muestra el programa para el caso MLG Bernoulli.

```
n <- 3 #Número de puntos a elegir en la región truncada
N <<- 10000#Largo de la cadena
p <<- 2 #Número de parámetros
B <<- rbind(c(-1,0),c(0,-1),c(-1,1),c(1,4), c(2,1), c(3,-4)) #matriz B
b <<- array(c(0,0,5,40,27,24),c(nrow(B),1))# vector b en la desigualdad
quitaB <<- function(n){matrix(B[,-(c(n))], nrow=nrow(B), ncol=ncol(B)-1)}
quitabetas <<- function(n,z){as.vector(z[-n])}
mu <- array(0,c(p,1))
sigma <- diag(100,p)
library(MASS)
library(msm)
A <<- solve(t(chol(sigma)))
Ainv <<- t(chol(sigma))
DD <<- B%*%Ainv
alfaz <<- A%*%mu
quitaD <<- function(n){matrix(DD[,-(c(n))], nrow=nrow(DD), ncol=ncol(DD)-1)}
quitaz <<- function(n,z){as.vector(z[-n])}
b0 <- ginv(B)%*%b
z0 <- array(A%*%b0, c(p,1))
muestrasyam<-matrix(nrow=N,ncol=p)
muestraszetas<-array(rep(0,p),c(p,1))
for(l in 1:N)
{ limi <- array(c(p,1))
  lims <- array(c(p,1))
  for(j in 1:p)
  { limii <- array(rep(-Inf,length(b)),c(length(b),1))
    limss <- array(rep(Inf,length(b)),c(length(b),1))
    U <- (b - quitaD(j)%*%quitaz(j,z0))
    V <- DD[,j]
    for(i in 1:length(V))
    {if(V[i]!=0)
      { if(V[i]>0) limss[i] <- U[i]/V[i]
        else limii[i] <- U[i]/V[i]
      }
    }
  }
  if (max(limii) < min(limss))
  { limi[j] <- max(limii)
    lims[j] <- min(limss)
    muestraszetas[j] <- rtnorm(1,mean=alfaz[j],sd=1,lower=limi[j],upper=lims[j])
    z0[j] <- muestraszetas[j]
  }
}
muestrasyam[1,] <- Ainv%*%muestraszetas
z0 <- muestraszetas
}
```

```

#Toma n puntos de la cadena, con saltos N/n
betasaletorios <- matrix(nrow=n,ncol=p)
saltos <- N/n
for(i in 1:n)
  { betasaletorios[i,] <- muestrasyam[saltos,]
    saltos <- saltos + (N/n)
  }

#Prueba para verificar condición en la región de truncada
w <- 0
for(i in 1:nrow(betasaletorios))
  { d <- B%%betasaletorios[i,]
    e <- 0
    for(j in 1:length(d))
      {if(d[j] < b[j] | d[j]==b[j]) {e <- e + 1}}
    if(e==length(d)) {w <- w + 1}
    else cat('No cumple posición:', '\n', i, '\n')
  }

cat('Número de vectores que cumplen la condición de truncamiento:', '\n')
w

#Grafica los n puntos elegidos en el soporte T
x <- c(0,0, 4, 68/7, 12, 8)
y <- c(0, 5, 9, 53/7, 3, 0)
plot(x,y, xlab=expression(beta[1]),ylab=expression(beta[2]),pch=19, col="white")
polygon(x,y, col = "lightgray", lty=1, lwd=1, border="black")
for(i in 1:n) {points(betasaletorios[i,1],betasaletorios[i,2],
  pch=20,col = "black", cex = 1)}
text(-0.1,0.3,labels="(0,0)")
text(-0.1,5.3,labels="(0,5)")
text(3.3,9,labels="(4,9)")
text((68/7)+.3, (53/7)+.3,labels="(68/7,53/7)")
text(11.7,2,labels="(12,3)")
text(9,0.3,labels="(8,0)")

save(betasaletorios,file="C:/Simulados/Bernoulli/PuntosinicialesBer.Rdata")

```

A3. Implementación del GTMI

En este anexo se muestra el programa escrito en lenguaje R (R Development Core Team, 2007), el cual implementa el GTMI cuando la distribución a priori del vector de parámetros es una Uniforme. Se muestra el caso MLG Bernoulli.

```
#MGTMI. Bernoulli, caso 1 cuando beta es (4,3) y n=30
library(msm)
k <- 10000 #Largo de la cadena
n <- 30 #tamaño de la muestra
p <- 2 #Número de betas
B <- rbind(c(-1,0),c(0,-1),c(-1,1),c(1,4), c(2,1), c(3,-4)) #matriz B
b <- array(c(0,0,5,40,27,24),c(nrow(B),1))# vector b en la desigualdad
#Lee los datos de un archivo de texto
load("C:/Simulados/Bernoulli/Datos1Ber30.Rdata")
#Aproximación de la función aposterior para beta
moda0 <- array(c(0,0),c(p,1))#valor inicial
error <- cbind(rep(0.1,p))
z <- c(rep(0,n))
invsigmaz <- matrix(0,ncol=n, nrow=n)
while(sum(error > 0.0001) != 0)
{ for(i in 1:n)
  { d <- t(X[i,])%*%moda0
    e <- exp(d)
    z[i] <- d + (((1+e)^2/e)*(y[i] - (e/(1+e))))
    invsigmaz[i,i] <- e/(1+e)^2
  }
  sigmam <- solve(t(X)%*%invsigmaz%*%X)
  betam <- sigmam%*%t(X)%*%invsigmaz%*%z
  error[1] <- abs((betam[1]-moda0[1])/moda0[1])
  error[2] <- abs((betam[2]-moda0[2])/moda0[2])
  moda0 <- betam
}
cat('Por lo tanto, la media de la distribución aproximada es:', '\n')
betam
cat('Y la matriz de covarianzas es:', '\n')
sigmam
#Función Log-aposteriori aproximada (Normal multivariada truncada)
betaposterap <- function(b1)
  {return((((-1/2)*t(b1-betam)%*%solve(sigmam)%*(b1-betam))[1,1])}
#Muestreando la aposteriori aproximada(Gibbs Sampler de Rodríguez Yam)
A <- solve(t(chol(sigmam)))#Matriz de rango completo, tal que A%*%sigmam%*%t(A)=I
Ainv <- t(chol(sigmam))
DD <- B%*%Ainv
alfaz <- A%*%betam
quitaD <- function(n){matrix(DD[,-(c(n))], nrow=nrow(DD), ncol=ncol(DD)-1)}
quitaz <- function(n,z){as.vector(z[-n])}
#Implementación
load("C:/Simulados/Bernoulli/PuntosinicialesBer.Rdata")
```

```

#Cadena 1
cadena1MGMTMI <- matrix(nrow=k,ncol=p)
b0 <- betasaletorios[1,]
z0 <- array(A%*%b0, c(p,1))
muestraszetas<-array(rep(0,p),c(p,1))
for(l in 1:k)
  {limi <- array(c(p,1))
  lims <- array(c(p,1))
  for(j in 1:p)
    {limii <- array(rep(-Inf,length(b)),c(length(b),1))
    limss <- array(rep(Inf,length(b)),c(length(b),1))
    U <- (b - quitaD(j)%*%quitaz(j,z0))
    V <- DD[,j]
    for(i in 1:length(V))
      {if(V[i]!=0)
      {if(V[i]>0) limss[i] <- U[i]/V[i]
      else limii[i] <- U[i]/V[i]
      }
      }
    if(max(limii) < min(limss))
      {limi[j] <- max(limii)
      lims[j] <- min(limss)
      muestraszetas[j] <- rtnorm(1,mean=alfaz[j],sd=1,lower=limi[j],upper=lims[j])
      z0[j] <- muestraszetas[j]
      }
    }
  }
  cadena1MGMTMI[l,] <- array(Ainv%*%muestraszetas,c(1,p))
  z0 <- muestraszetas
}

#Cadena 2
cadena2MGMTMI <- matrix(nrow=k,ncol=p)
b0 <- betasaletorios[2,]
z0 <- array(A%*%b0, c(p,1))
muestraszetas<-array(rep(0,p),c(p,1))
for(l in 1:k)
  {limi <- array(c(p,1))
  lims <- array(c(p,1))
  for(j in 1:p)
    {limii <- array(rep(-Inf,length(b)),c(length(b),1))
    limss <- array(rep(Inf,length(b)),c(length(b),1))
    U <- (b - quitaD(j)%*%quitaz(j,z0))
    V <- DD[,j]
    for(i in 1:length(V))
      {if(V[i]!=0)
      {if(V[i]>0) limss[i] <- U[i]/V[i]
      else limii[i] <- U[i]/V[i]
      }
      }
    if(max(limii) < min(limss))
      {limi[j] <- max(limii)

```

```

        lims[j] <- min(limss)
        muestraszetas[j] <- rtnorm(1,mean=alfaz[j],sd=1,lower=limi[j],upper=lims[j])
        z0[j] <- muestraszetas[j]
    }
}
cadena2MGMTMI[l,] <- array(Ainv%*%muestraszetas,c(1,p))
z0 <- muestraszetas
}
#Cadena 3
cadena3MGMTMI <- matrix(nrow=k,ncol=p)
b0 <- betasaleatorios[3,]
z0 <- array(A%*%b0, c(p,1))
muestraszetas<-array(rep(0,p),c(p,1))
for(l in 1:k)
{limi <- array(c(p,1))
  lims <- array(c(p,1))
  for(j in 1:p)
  {limii <- array(rep(-Inf,length(b)),c(length(b),1))
    limss <- array(rep(Inf,length(b)),c(length(b),1))
    U <- (b - quitaD(j)%*%quitaz(j,z0))
    V <- DD[,j]
    for(i in 1:length(V))
    {if(V[i]!=0)
      {if(V[i]>0) limss[i] <- U[i]/V[i]
        else limii[i] <- U[i]/V[i]
      }
    }
    if(max(limii) < min(limss))
    {limi[j] <- max(limii)
      lims[j] <- min(limss)
      muestraszetas[j] <- rtnorm(1,mean=alfaz[j],sd=1,lower=limi[j],upper=lims[j])
      z0[j] <- muestraszetas[j]
    }
  }
}
cadena3MGMTMI[l,] <- array(Ainv%*%muestraszetas,c(1,p))
z0 <- muestraszetas
}
#Salvando datos
save(betam,sigmam,file="C:/Simulados/Bernoulli/NoInformativa/
  Berbetamsigmam30-1MGMTMI.Rdata")
save(cadena1MGMTMI,file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGMTMI1-30.Rdata")
save(cadena2MGMTMI,file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGMTMI1-30.Rdata")
save(cadena3MGMTMI,file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGMTMI1-30.Rdata")

```

A4. Implementación del GDS

En este anexo se muestra el programa escrito en lenguaje R (R Development Core Team, 2007), el cual implementa el GDS cuando la distribución a priori del vector de parámetros es una Uniforme. Se muestra el caso MLG Bernoulli. La parte del programa que implementa el algoritmo ARS fue escrito por Michael Höhle del departamento de estadística de la Universidad de Munich.

```
#MGDS. Bernoulli, caso 1 cuando beta es (4,3) y n=30
k <- 10000 #Largo de la cadena
n <- 30 #tamaño de la muestra
p <- 2 #Número de betas
B <- rbind(c(-1,0),c(0,-1),c(-1,1),c(1,4), c(2,1), c(3,-4)) #matriz B
b <- array(c(0,0,5,40,27,24),c(nrow(B),1))#vector b en la desigualdad
quitaB <- function(n){matrix(B[,-(c(n))], nrow=nrow(B), ncol=ncol(B)-1)}
quitabeta <- function(n,z){as.vector(z[-n])}
#Lee los datos de un archivo de texto
load("C:/Simulados/Bernoulli/Datos1Ber30.Rdata")
#La función Log-Verosimilitud
verosimi <- function(b1)
  {b11 <- array(b1,c(p,1))
  suma1 <- 0
  suma2 <- 0
  for(i in 1:n)
    { a <- X[i,]%*%b11
      suma1 <- suma1 + (y[i]*a)
      suma2 <- suma2 + log(1+exp(a))
    }
  return((suma1 - suma2)[1,1])
}
#Definiendo las full-condicionales
f <- function(x) { b0[j] <- x
  return(exp(verosimi(b0)))
}

h <- function(x){log(f(x))}
#Usa el método ARS para muestrear las full condicionales
int.intervall <- function()
{int.one <- function(i)
  {exp(coeffE[2,i])/coeffE[1,i]*(exp(coeffE[1,i]*z[i+1])-exp(coeffE[1,i]*z[i]))}
  sapply(1:(length(z)-1),int.one)
}
setup.ars <- function(S)
  { noOfPoints <- length(S)
  H <- array(c(noOfPoints,1))
  for(l in 1:noOfPoints)
    {H[l] <- h(S[l])}
  L <- matrix(1e99,nrow=2,ncol=length(S))
  for (i in 1:(length(S)-1))
    { L[1,i] <- (H[i+1]-H[i])/(S[i+1]-S[i])
      L[2,i] <- H[i]-L[1,i]*S[i]
    }
}
```

```

if (noOfPoints>5)
  { z <- NULL
    z <- S[1:3]
    for(i in 3:(noOfPoints-3))
      {z <- c(z,-(L[2,i+1]-L[2,i-1])/(L[1,i+1]-L[1,i-1]),S[i+1])}
      z <- c(z,S[(noOfPoints-1):noOfPoints])
    } else {z <- S}
coeffE <- matrix(NA,2,length(z))
coeffE[,1] <- L[,2]
coeffE[,2] <- L[,3]
i <- j <- 3
while (i < (noOfPoints-2))
  { coeffE[,j] <- L[,i-1]
    coeffE[,j+1] <- L[,i+1]
    j <- j+2
    i <- i+1
  }
coeffE[,j] <- L[,i-1]
coeffE[,j+1] <- L[,i]
S <<- S
noOfPoints <<- length(S)
H <<- H
L <<- L
z <<- z
coeffE <<- coeffE
weights <- c(0,int.intervall())
norm <- 1/sum(weights)
weights <- weights/sum(weights)
csumw <- cumsum(weights)
csumw <<- csumw
norm <<- norm
}
squeeze <- function(x)
  {n <- length(S)
  one <- function(x)
    { idx <- which.min(x > S) - 1
      if (idx == 1 | idx == (n-1)) {return(-Inf)} else
        {return(L[2,idx] + L[1,idx]*x)}
    }
  sapply(x,one)
}
envelope <- function(x)
  { one <- function(x)
    { idx <- which.min(x > z) - 1
      return(coeffE[1,idx]*x + coeffE[2,idx])
    }
  sapply(x,one)
}
renvelope <- function(size)
{u <- runif(size)

```

```

i <- sapply(u,function(u) which.max(csumw>u) - 1)
umark <- u-csumw[i]
x <- (log(exp(coeffE[2,i]+coeffE[1,i]*z[i])+umark*coeffE[1,i]/norm)
      -coeffE[2,i])/coeffE[1,i]
return(x)
}
ars <- function(maxPoints=20,plot=F,adaptive=F)
{while(TRUE){x<-renvelope(1)
  u <- runif(1)
  if (u <= exp(squeeze(x)-envelope(x))) {return(x)} else
  {if (u <=f(x)/exp(envelope(x))){if((noOfPoints< maxPoints)&
    (adaptive)){S<<-sort(c(x,S))
      setup.ars(S)
    }
    return(x)}
  }
}
}
Implementación MGDS
load("C:/Simulados/Bernoulli/PuntosinicialesBer.Rdata")
#Cadena 1
b0 <- betasaletorios[1,]
cadena1MGDS <- matrix(nrow=k,ncol=p)
for(l in 1:k)
{limi <- array(c(p,1))
  lims <- array(c(p,1))
  for(j in 1:p)
  {limii <- array(rep(-Inf,length(b)),c(length(b),1))
    limss <- array(rep(Inf,length(b)),c(length(b),1))
    U <- (b - quitaB(j)%*%quitabeta(j,b0))
    V <- B[,j]
    for(i in 1:length(V))
    {if(V[i]!=0)
      {if(V[i]>0) limss[i] <- U[i]/V[i]
        else limii[i] <- U[i]/V[i]
      }
    }
  }
  if (max(limii) < min(limss))#si hay solución en el sistema, entonces
  {limi[j] <- max(limii)
    lims[j] <- min(limss)
    if(limi[j] != -Inf && lims[j] != Inf)
    {S <- c((7*limi[j]+lims[j])/8,(3*limi[j]+lims[j])/4,(5*limi[j]+3*lims[j])/8,
      (limi[j]+lims[j])/2,(5*lims[j]+3*limi[j])/8,(3*lims[j]+limi[j])/4,
      (7*lims[j]+limi[j])/8)
      setup.ars(S)
      cadena1MGDS[l,j] <- replicate(1,ars(maxPoints=20,adaptive=F))
      b0[j] <- cadena1MGDS[l,j]
    }
  }
}
}
}

```

```

    b0 <- cadena1MGDS[1,]
  }
#Cadena 2
b0 <- betasaletorios[2,]
cadena2MGDS <- matrix(nrow=k,ncol=p)
for(l in 1:k)
{limi <- array(c(p,1))
  lims <- array(c(p,1))
  for(j in 1:p)
  {limii <- array(rep(-Inf,length(b)),c(length(b),1))
    limss <- array(rep(Inf,length(b)),c(length(b),1))
    U <- (b - quitaB(j)%*%quitabeta(j,b0))
    V <- B[,j]
    for(i in 1:length(V))
    {if(V[i]!=0)
      {if(V[i]>0) limss[i] <- U[i]/V[i]
        else limii[i] <- U[i]/V[i]
      }
    }
    if (max(limii) < min(limss))#si hay solución en el sistema, entonces
    {limi[j] <- max(limii)
      lims[j] <- min(limss)
      if(limi[j] != -Inf && lims[j] != Inf)
      {S <- c((7*limi[j]+lims[j])/8,(3*limi[j]+lims[j])/4,(5*limi[j]+3*lims[j])/8,
        (limi[j]+lims[j])/2,(5*lims[j]+3*limi[j])/8,(3*lims[j]+limi[j])/4,
        (7*lims[j]+limi[j])/8)
        setup.ars(S)
        cadena2MGDS[1,j] <- replicate(1,ars(maxPoints=20,adaptive=F))
        b0[j] <- cadena2MGDS[1,j]
      }
    }
  }
}
b0 <- cadena2MGDS[1,]
}
#Cadena 3
b0 <- betasaletorios[3,]
cadena3MGDS <- matrix(nrow=k,ncol=p)
for(l in 1:k)
{limi <- array(c(p,1))
  lims <- array(c(p,1))
  for(j in 1:p)
  {limii <- array(rep(-Inf,length(b)),c(length(b),1))
    limss <- array(rep(Inf,length(b)),c(length(b),1))
    U <- (b - quitaB(j)%*%quitabeta(j,b0))
    V <- B[,j]
    for(i in 1:length(V))
    {if(V[i]!=0)
      {if(V[i]>0) limss[i] <- U[i]/V[i]
        else limii[i] <- U[i]/V[i]
      }
    }
  }
}

```

```

}
if (max(limii) < min(limss))#si hay solución en el sistema, entonces
{limi[j] <- max(limii)
  lims[j] <- min(limss)
  if(limi[j] != -Inf && lims[j] != Inf)
    {S <- c((7*limi[j]+lims[j])/8,(3*limi[j]+lims[j])/4,(5*limi[j]+3*lims[j])/8,
            (limi[j]+lims[j])/2,(5*lims[j]+3*limi[j])/8,(3*lims[j]+limi[j])/4,
            (7*lims[j]+limi[j])/8)
      setup.ars(S)
      cadena3MGDS[1,j] <- replicate(1,ars(maxPoints=20,adaptive=F))
      b0[j] <- cadena3MGDS[1,j]
    }
  }
}
b0 <- cadena3MGDS[1,]
}
#Salvando datos
save(cadena1MGDS,file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGDS1-30.Rdata")
save(cadena2MGDS,file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGDS1-30.Rdata")
save(cadena3MGDS,file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGDS1-30.Rdata")

```

A5. Gráficas

En este anexo se muestra el programa escrito en lenguaje R (R Development Core Team, 2007), el cual permite obtener todas las gráficas mostradas en este trabajo, el programa ilustra el caso MLG Bernoulli.

```
#Gráfica de la región de truncamiento y los valores del parámetro. Caso Bernoulli
x <- c(0,0, 4, 68/7, 12, 8)
y <- c(0, 5, 9, 53/7, 3, 0)
plot(x,y, xlab=expression(beta[1]),ylab=expression(beta[2]),pch=19,col="white")
polygon(x,y, col = "lightgray", lty=1, lwd=1, border="black")
points(4,3,pch=19,col = "black", cex = 1.2)
points(4.5,8.3,pch=19,col = "black", cex = 1.2)
points(6.2,0.8,pch=19,col = "black", cex = 1.2)
text(4.3,3.3,labels=expression(paste(beta," = (4 , 3)")))
text(4.8,8,labels=expression(paste(beta," = (4.5 , 8.3)")))
text(6.5,1.1,labels=expression(paste(beta," = (6.2 , 0.8)")))
text(-0.1,0.3,labels="(0,0)")
text(-0.1,5.3,labels="(0,5)")
text(3.3,9,labels="(4,9)")
text((68/7)+.3,(53/7)+.3,labels="(68/7,53/7)")
text(11.7,2,labels="(12,3)")
text(9,0.1,labels="(8,0)")
#####
#Bernoulli caso 1, beta es (4,3), n=30
#Gráfica de la distribución final y aproximada para beta, así como la de contornos.
load("C:/Simulados/Bernoulli/Datos1Ber30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Berbetamsigam30-1MGTM.I.Rdata")
p <- 2#núm. de parámetros
n <- 30
#La función Verosimilitud
verosimi <- function(b1)
  {b11 <- array(b1,c(p,1))
  suma1 <- 0
  suma2 <- 0
  for(i in 1:n)
    { a <- X[i,]%*%b11
      suma1 <- suma1 + (y[i]*a)
      suma2 <- suma2 + log(1+exp(a))
    }
  return(exp((suma1 - suma2)[1,1]))
}
betaprior <- function(b3){return(1)}
#Función aposteriori verdadera para beta
const <- ((2*pi)^(p/2))*(sqrt(det(sigam)))*verosimi(betam)*betaprior(betam)
betaposter <-function(b3){return((1/const)*verosimi(b3)*betaprior(b3))}
#Función aposteriori aproximada (Normal multivariada truncada)
betaposterap <- function(b1)
  {return((sqrt(det(solve(sigam)))/(2*pi))*exp(((1/2)
    *t(b1-betam)%*%solve(sigam)%*(b1-betam))[1,1]))}
#Definiendo el grid
nx <- 50
```

```

beta1 <- array(seq(0,11, length = nx),c(nx,1))
beta2 <- array(seq(0,6, length = nx),c(nx,1))
B <-< rbind(c(-1,0),c(0,-1),c(-1,1),c(1,4), c(2,1), c(3,-4))
b <-< array(c(0,0,5,40,27,24),c(nrow(B),1))
w1 <- matrix(0,nrow=nx, ncol=nx)
for(i in 1:nx)
  { for(k in 1:nx)
    { d <- B%*%c(beta1[i],beta2[k])
      e <- 0
      for(j in 1:length(d))
        {if(d[j] < b[j] | d[j]==b[j]) {e <- e + 1}}
      if(e==length(d))
        {w1[i,k] <- betaposter(c(beta1[i],beta2[k]))}
    }
  }
}
#Evaluación en el grid de la aposteriori aproximada
w2 <- matrix(0,nrow=nx, ncol=nx)
for(i in 1:nx)
  { for(k in 1:nx)
    { d <- B%*%c(beta1[i],beta2[k])
      e <- 0
      for(j in 1:length(d))
        {if(d[j] < b[j] | d[j]==b[j]) {e <- e + 1}}
      if(e==length(d))
        {w2[i,k] <- betaposterap(c(beta1[i],beta2[k]))}
    }
  }
}
#Gráfica de las funciones aposteriori verdadera y aproximada para beta
X11()
par(mfrow=c(1,2))
persp(beta1, beta2, w1,theta = 45, phi = 15, expand = 0.5,
  col = "lightblue",ltheta = 120,lphi = 60, shade = 0.1,
  ticktype = "detailed",xlab="", ylab="", zlab="")
persp(beta1, beta2, w2,theta = 45, phi = 15, expand = 0.5,
  col = "lightblue",ltheta = 120,lphi = 60, shade = 0.1,
  ticktype = "detailed",xlab="", ylab="", zlab="")
X11()
par(mfrow=c(1,2))
contour(beta1,beta2,w1,lwd=1.5,nlevels=5, xlab=expression(beta[1]),
  ylab=expression(beta[2]))
contour(beta1,beta2,w2,nlevels=5,labels=c(0.01,0.1,0.5),lwd=1.5,
  xlab=expression(beta[1]),ylab=expression(beta[2]))
#####
#Gráficas de las cadenas-Bernoulli, tamaño n=30. Caso 1
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGTM1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGTM1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGTM1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGDS1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGDS1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGDS1-30.Rdata")

```

```

load(file="C:/Simulados/Bernoulli/PuntosinicialesBer.Rdata")
l <- 500 #Primeras l iteraciones a graficar
cadena1MGDMI <- cadena1MGDMI[(c(1:l)),]
cadena2MGDMI <- cadena2MGDMI[(c(1:l)),]
cadena3MGDMI <- cadena3MGDMI[(c(1:l)),]
cadena1MGDS <- cadena1MGDS[(c(1:l)),]
cadena2MGDS <- cadena2MGDS[(c(1:l)),]
cadena3MGDS <- cadena3MGDS[(c(1:l)),]
# Método-MGDMI
x11()
layout(matrix(c(1,2,3,4,5,6),byrow=T,ncol=3),heights=c(1,1,1),respect=T)
x <- c(0,0, 4, 68/7, 12, 8)
y <- c(0, 5, 9, 53/7, 3, 0)
plot(x,y, xlab=expression(beta[1]),ylab=expression(beta[2]),pch=19,
      col="white")
polygon(x,y, col = "lightgray", lty=1, lwd=1, border="black")
points(betasaletorios[1,1],betasaletorios[1,2],pch=17, col = "black",cex = 1.3)
for(i in 1:nrow(cadena1MGDMI)) {points(cadena1MGDMI[i,1],cadena1MGDMI[i,2],
                                       pch=19, col = "black", cex = 0.3)}
mtext("Cadena 1-GDMI",side=3,line=1, font=1)
x <- c(0,0, 4, 68/7, 12, 8)
y <- c(0, 5, 9, 53/7, 3, 0)
plot(x,y,xlab=expression(beta[1]),ylab=expression(beta[2]),pch=19,col="white")
polygon(x,y, col = "lightgray", lty=1, lwd=1, border="black")
points(betasaletorios[2,1],betasaletorios[2,2],pch=17, col = "black",cex = 1.3)
for(i in 1:nrow(cadena1MGDMI)) {points(cadena2MGDMI[i,1],cadena2MGDMI[i,2],
                                       pch=19, col = "black", cex = 0.3)}
mtext("Cadena 2-GDMI",side=3,line=1, font=1)
x <- c(0,0, 4, 68/7, 12, 8)
y <- c(0, 5, 9, 53/7, 3, 0)
plot(x,y,xlab=expression(beta[1]),ylab=expression(beta[2]),pch=19,col="white")
polygon(x,y, col = "lightgray", lty=1, lwd=1, border="black")
points(betasaletorios[3,1],betasaletorios[3,2],pch=17, col = "black",cex = 1.3)
for(i in 1:nrow(cadena1MGDMI)) {points(cadena3MGDMI[i,1],cadena3MGDMI[i,2],
                                       pch=19, col = "black", cex = 0.3)}
mtext("Cadena 3-GDMI",side=3,line=1, font=1)
# Método-MGDS
x <- c(0,0, 4, 68/7, 12, 8)
y <- c(0, 5, 9, 53/7, 3, 0)
plot(x,y, xlab=expression(beta[1]),ylab=expression(beta[2]),pch=19,col="white")
polygon(x,y, col = "lightgray", lty=1, lwd=1, border="black")
points(betasaletorios[1,1],betasaletorios[1,2],pch=17, col = "black",cex = 1.3)
for(i in 1:nrow(cadena1MGDS)) {points(cadena1MGDS[i,1],cadena1MGDS[i,2],
                                       pch=19, col = "black", cex = 0.3)}
mtext("Cadena 1-GDS",side=3,line=1, font=1)
x <- c(0,0, 4, 68/7, 12, 8)
y <- c(0, 5, 9, 53/7, 3, 0)
plot(x,y, xlab=expression(beta[1]),ylab=expression(beta[2]),pch=19, col="white")
polygon(x,y, col = "lightgray", lty=1, lwd=1, border="black")
points(betasaletorios[2,1],betasaletorios[2,2],pch=17, col = "black", cex = 1.3)

```

```

for(i in 1:nrow(cadena1MGDS)) {points(cadena2MGDS[i,1],cadena2MGDS[i,2],
                                     pch=19, col = "black", cex = 0.3)}
mtext("Cadena 2-GDS",side=3,line=1, font=1)
x <- c(0,0, 4, 68/7, 12, 8)
y <- c(0, 5, 9, 53/7, 3, 0)
plot(x,y, xlab=expression(beta[1]),ylab=expression(beta[2]),pch=19, col="white")
polygon(x,y, col = "lightgray", lty=1, lwd=1, border="black")
points(betasaletorios[3,1],betasaletorios[3,2],pch=17, col = "black", cex = 1.3)
for(i in 1:nrow(cadena1MGDS)) {points(cadena3MGDS[i,1],cadena3MGDS[i,2],
                                     pch=19, col = "black", cex = 0.3)}
mtext("Cadena 3-GDS",side=3,line=1, font=1)
#####
#Factor potencial de escala para el caso (4,3), Bernoulli, n=30. Según el MGTMI.
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGTMI1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGTMI1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGTMI1-30.Rdata")
cadena1MGTMI <- cadena1MGTMI[(c(1:1000)),]
cadena2MGTMI <- cadena2MGTMI[(c(1:1000)),]
cadena3MGTMI <- cadena3MGTMI[(c(1:1000)),]
v <- 20 #Cada v tamaño de muestra será calculado el factor de reducción
w <- 3#Num. de cadenas
#Para beta 1
muestrasMGTMI1 <- cbind(cadena1MGTMI[,1],cadena2MGTMI[,1],cadena3MGTMI[,1])
k <- nrow(muestrasMGTMI1)
m <<- k/v #Núm. de particiones
iteraciones <- array(seq(v,k,v),c(m,1))
ybarsMGTMI <- matrix(nrow=w,ncol=m)
sMGTMI <- matrix(nrow=w,ncol=m)
for(r in 1:m)
  { s1 <- r*v
    for(l in 1:w)
      { ybarsMGTMI[l,r] <- mean(muestrasMGTMI1[1:s1,l])
        sMGTMI[l,r] <- var(muestrasMGTMI1[1:s1,l])
      }
  }
WWMGTMI1 <- array(c(m,1))
BBMGTMI1 <- array(c(m,1))
RMGTMI1 <- array(c(m,1))
for(i in 1:m)
  { WWMGTMI1[i] <- mean(sMGTMI[,i])
    BBMGTMI1[i] <- (var(ybarsMGTMI[,i]))*iteraciones[i]
    RMGTMI1[i] <- sqrt((((iteraciones[i]-1)/iteraciones[i])*WWMGTMI1[i]
                       + ((1/iteraciones[i])*BBMGTMI1[i]))/WWMGTMI1[i])
  }
#Para beta 2
muestrasMGTMI2 <- cbind(cadena1MGTMI[,2],cadena2MGTMI[,2],cadena3MGTMI[,2])
ybarsMGTMI <- matrix(nrow=w,ncol=m)
sMGTMI <- matrix(nrow=w,ncol=m)
for(r in 1:m)
  { s1 <- r*v

```

```

        for(l in 1:w)
            { ybarsMGMTMI[l,r] <- mean(muestrasMGMTMI2[1:s1,l])
              sMGMTMI[l,r] <- var(muestrasMGMTMI2[1:s1,l])
            }
        }
WWMGMTMI2 <- array(c(m,1))
BBMGMTMI2 <- array(c(m,1))
RMGMTMI2 <- array(c(m,1))
for(i in 1:m)
{ WWMGMTMI2[i] <- mean(sMGMTMI[,i])
  BBMGMTMI2[i] <- (var(ybarsMGMTMI[,i]))*iteraciones[i]
  RMGMTMI2[i] <- sqrt((((iteraciones[i]-1)/iteraciones[i])*WWMGMTMI2[i]
                      + ((1/iteraciones[i])*BBMGMTMI2[i]))/WWMGMTMI2[i])
}
#Factor potencial de escala para el caso (4,3), Bernoulli, n=30. Según el MGDS.
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGDS1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGDS1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGDS1-30.Rdata")
#Para beta 1
muestrasMGDS1 <- cbind(cadena1MGDS[,1],cadena2MGDS[,1],cadena3MGDS[,1])
ybarsMGDS <- matrix(nrow=w,ncol=m)
sMGDS <- matrix(nrow=w,ncol=m)
for(r in 1:m)
{ s1 <- r*v
  for(l in 1:w)
    { ybarsMGDS[l,r] <- mean(muestrasMGDS1[1:s1,l])
      sMGDS[l,r] <- var(muestrasMGDS1[1:s1,l])
    }
  }
WWMGDS1 <- array(c(m,1))
BBMGDS1 <- array(c(m,1))
RMGDS1 <- array(c(m,1))
for(i in 1:m)
{ WWMGDS1[i] <- mean(sMGDS[,i])
  BBMGDS1[i] <- (var(ybarsMGDS[,i]))*iteraciones[i]
  RMGDS1[i] <- sqrt((((iteraciones[i]-1)/iteraciones[i])*WWMGDS1[i]
                    + ((1/iteraciones[i])*BBMGDS1[i]))/WWMGDS1[i])
}
#Para beta 2
muestrasMGDS2 <- cbind(cadena1MGDS[,2],cadena2MGDS[,2],cadena3MGDS[,2])
ybarsMGDS <- matrix(nrow=w,ncol=m)
sMGDS <- matrix(nrow=w,ncol=m)
for(r in 1:m)
{ s1 <- r*v
  for(l in 1:w)
    { ybarsMGDS[l,r] <- mean(muestrasMGDS2[1:s1,l])
      sMGDS[l,r] <- var(muestrasMGDS2[1:s1,l])
    }
  }
}
WWMGDS2 <- array(c(m,1))

```

```

BBMGDS2 <- array(c(m,1))
RMGDS2 <- array(c(m,1))
for(i in 1:m)
{ WWMGDS2[i] <- mean(sMGDS[,i])
  BBMGDS2[i] <- (var(ybarsMGDS[,i]))*iteraciones[i]
  RMGDS2[i] <- sqrt((((iteraciones[i]-1)/iteraciones[i])*WWMGDS2[i]
                    + ((1/iteraciones[i])*BBMGDS2[i]))/WWMGDS2[i])
}
#Gráfica del factor de escala por método para cada componente
x11()
par(mfrow=c(2,1))
plot(iteraciones,RMGDS1,type="l",lwd=1,lty=1,xlab="Iteraciones",
      ylab=expression(paste(hat(R)," para ",beta[1])),cex=1.5)
lines(iteraciones,RMGDMI1,type="l",lwd=1,lty=2,cex=1.5)
abline(h = 1, col = "black", lty=1)
legend("topright", legend = c("MGDS", "MGDMI"),lty = c(1,2),
       xjust = 1, yjust = 1,lwd=1,bty="n")
plot(iteraciones,RMGDS2,type="l",lwd=1,lty=1,xlab="Iteraciones",
      ylab=expression(paste(hat(R)," para ",beta[2])),cex=1.5)
lines(iteraciones,RMGDMI2,type="l",lwd=1,lty=2,cex=1.5)
abline(h = 1, col = "black", lty=1)
legend("topright", legend = c("MGDS", "MGDMI"),lty = c(1,2),
       xjust = 1, yjust = 1,lwd=1,bty="n")
#####
# Gráficas para el Diagnóstico de convergencia-Bernoulli Caso 1 tamaño n=30
library(coda)#abre la librería coda
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGDMI1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGDMI1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGDMI1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGDS1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGDS1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGDS1-30.Rdata")
#Se usan las segundas mitades de las cadenas de cada método para generar
#la matriz "cadenas" por método en formato mcmc
cadenasMGDMI <- mcmc.list(list(mcmc(cadena1MGDMI,5001,nrow(cadena1MGDMI)),
                             mcmc(cadena2MGDMI,5001,nrow(cadena2MGDMI)),
                             mcmc(cadena3MGDMI,5001,nrow(cadena3MGDMI))))
cadenasMGDS <- mcmc.list(list(mcmc(cadena1MGDS,5001,nrow(cadena1MGDS)),
                             mcmc(cadena2MGDS,5001,nrow(cadena2MGDS)),
                             mcmc(cadena3MGDS,5001,nrow(cadena3MGDS))))
#Se grafican las autocorrelaciones por método
x11()
nf <- layout(rbind(c(1,1,2,2),c(3,3,4,4),c(5,5,6,6)))
layout.show(nf)
autocorr.plot(cadenasMGDMI,auto.layout=F)
x11()
nf <- layout(rbind(c(1,1,2,2),c(3,3,4,4),c(5,5,6,6)))
layout.show(nf)
autocorr.plot(cadenasMGDS,auto.layout=F)

```

```

#Muestra la gráfica de la evolución del factor de escala conforme
el número de iteraciones crece
titulos <- list(expression(beta[1]),expression(beta[2]))
x11()
nf <- layout(rbind(c(1,1,3,3),c(2,2,4,4)))
layout.show(nf)
gelman.plot(cadenasMGTMi,auto.layout=F, xlab="Iteraciones",
            ylab=expression(paste("MGTMi- ",hat(R))),titulos=titulos)
gelman.plot(cadenasMGDS,auto.layout=F, xlab="Iteraciones",
            ylab=expression(paste("MGDS- ",hat(R))),titulos=titulos)

#Se grafican las tres cadenas y las marginales por Método,
para cada componente del vector de parámetros. Se toma la segunda mitad
verdadero <- c(4,3)#valor verdadero del parámetro
p <- 2 #número de parámetros
cadena1MGTMi <- cadena1MGTMi[(c(5001:10000)),]
cadena2MGTMi <- cadena2MGTMi[(c(5001:10000)),]
cadena3MGTMi <- cadena3MGTMi[(c(5001:10000)),]
cadena1MGDS <- cadena1MGDS[(c(5001:10000)),]
cadena2MGDS <- cadena2MGDS[(c(5001:10000)),]
cadena3MGDS <- cadena3MGDS[(c(5001:10000)),]
for(j in 1:p)
{ x11()
  nf <- layout(rbind(c(1,2),c(3,3)))
  layout.show(nf)
  plot(cadena1MGTMi[,j],type="l",xlab=titulos[j],ylab="Variación-MGTMi",
       lwd=1,lty=1,cex=1.5,col="red")
  points(cadena2MGTMi[,j],type="l",lwd=1,lty=1,cex=1.5,col="green")
  points(cadena3MGTMi[,j],type="l",lwd=1,lty=1,cex=1.5,col="grey")
  abline(h = verdadero[j], col = "black", lty=3)
  plot(cadena1MGDS[,j],type="l",xlab=titulos[j],ylab="Variación-MGDS",
       lwd=1,lty=1,cex=1.5,col="red")
  points(cadena2MGDS[,j],type="l",lwd=1,lty=1,cex=1.5,col="green")
  points(cadena3MGDS[,j],type="l",lwd=1,lty=1,cex=1.5,col="grey")
  abline(h = verdadero[j], col = "black", lty=3)
  plot(density(cbind(cadena1MGTMi[,j],cadena2MGTMi[,j],cadena3MGTMi[,j])),
       type="l",main="",xlab=titulos[j],ylab="Densidad",lwd=1,lty=1,cex=1.5)
  abline(v = verdadero[j], col = "lightgray", lty=3)
  points(density(cbind(cadena1MGDS[,j],cadena2MGDS[,j],cadena3MGDS[,j])),
       type="l",main="",xlab="",ylab="",lwd=1,lty=5,cex=1.5)
  legend("topright", legend = c("MGTMi", "MGDS"),lty = c(1,5), xjust = 1,
       yjust = 1,lwd=1,bty="n")
}

```

A6. Estadísticas

En este anexo se muestra el programa escrito en lenguaje R (R Development Core Team, 2007), el cual permite obtener las estadísticas mostradas en este trabajo, el programa ilustra el caso MLG Bernoulli.

```
#Bernoulli caso 1. beta=c(4,3), tamaño n=30. Estimaciones a posteriori
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGMTMI1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGMTMI1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGMTMI1-30.Rdata")
load("C:/Simulados/Bernoulli/Datos1Ber30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Berbetamsigmam30-1MGMTMI.Rdata")
n <- 30#tamaño de muestra
verdadero <- c(4,3)
p <- 2#núm. de parámetros
m <- 3 #núm. de cadenas
#La función Verosimilitud
verosimi <- function(b1)
  {b11 <- array(b1,c(p,1))
  suma1 <- 0
  suma2 <- 0
  for(i in 1:n)
    { a <- X[i,]%*%b11
      suma1 <- suma1 + (y[i]*a)
      suma2 <- suma2 + log(1+exp(a))
    }
  return(exp((suma1 - suma2)[1,1]))
}
#Función a posteriori verdadera para beta
betaposter <-function(b3){return(verosimi(b3))}

#Función a posteriori aproximada (Normal multivariada truncada)
betaposterap <- function(b1)
  {return(exp(((1/2)*t(b1-betam)%*%solve(sigmam)%*(b1-betam))[1,1]))}
#Tomando la segunda mitad de cada cadena
muestrasbetas1MGMTMI <- cbind(cadena1MGMTMI[,1][-(1:5000)],cadena2MGMTMI[,1][-(1:5000)],
  cadena3MGMTMI[,1][-(1:5000)])
muestrasbetas2MGMTMI <- cbind(cadena1MGMTMI[,2][-(1:5000)],cadena2MGMTMI[,2][-(1:5000)],
  cadena3MGMTMI[,2][-(1:5000)])

#mezclando las cadenas
muestrasbetasMGMTMI <- matrix(nrow=m*nrow(muestrasbetas1MGMTMI),ncol=p)
muestrasbetasMGMTMI[,1] <- c(muestrasbetas1MGMTMI[,1],muestrasbetas1MGMTMI[,2],
  muestrasbetas1MGMTMI[,3])
muestrasbetasMGMTMI[,2] <- c(muestrasbetas2MGMTMI[,1],muestrasbetas2MGMTMI[,2],
  muestrasbetas2MGMTMI[,3])

#Inferencia para la esperanza de beta usando la técnica IS
h1 <-<- function(b1,n){return(b1[n])}
sumawi <- 0
sumaMGMTMI <- array(0,c(p,1))
for(i in 1:nrow(muestrasbetasMGMTMI))
  { wi <- betaposter(muestrasbetasMGMTMI[i,])/betaposterap(muestrasbetasMGMTMI[i,])
    sumawi <- sumawi + wi
```

```

    for(j in 1:p)
      {sumaMGMTMI[j] <- sumaMGMTMI[j] + wi*h1(muestrasbetasMGMTMI[i,],j)}
  }
Eh1MGMTMI <- array(c(p,1))#vector de medias
for(i in 1:p)
  {Eh1MGMTMI[i] <- sumaMGMTMI[i]/sumawi}

#Inferencia para la varianza de los componentes de beta usando la técnica IS
h2 <- function(b1,n){return((b1[n]-Eh1MGMTMI[n])^2)}
sumawi <- 0
sumaMGMTMI <- array(0,c(p,1))
for(i in 1:nrow(muestrasbetasMGMTMI))
  { wi <- betaposter(muestrasbetasMGMTMI[i,])/betaposterap(muestrasbetasMGMTMI[i,])
    sumawi <- sumawi + wi
    for(j in 1:p)
      {sumaMGMTMI[j] <- sumaMGMTMI[j] + wi*h2(muestrasbetasMGMTMI[i,],j)}
  }
Eh2MGMTMI <- array(c(p,1))#vector de varianzas para cada componente
for(i in 1:p)
  {Eh2MGMTMI[i] <- sumaMGMTMI[i]/sumawi}
sdbeta <- array(c(p,1))#vector de desviación estándar
for(i in 1:p)
  {sdbeta[i] <- sqrt(Eh2MGMTMI[i])}
sesgo <- array(c(p,1))#vector de sesgo
for(i in 1:p)
  {sesgo[i] <- verdadero[i]-Eh1MGMTMI[i]}
#Tabla Estadística según MGMTMI
table30MGMTMI<- matrix(0,p,3)
dimnames(table30MGMTMI)<-list(paste("beta",1:p,sep="_"),c("Media","Des. est.,"Sesgo"))
  for(i in 1:p)
    { table30MGMTMI[i,]<- c(Eh1MGMTMI[i],sdbeta[i],sesgo[i])}
#Guarda los resultados
save(table30MGMTMI,file="C:/Simulados/Bernoulli/NoInformativa/EstadisBer1MGMTMI-30.Rdata")
#####
#Resultados para el caso 1 cuando beta es (4,3). Bernoulli, n=30
load("C:/Simulados/Bernoulli/NoInformativa/Berbetamsigam30-1MGMTMI.Rdata")
cat('La media de la distribución aproximada para n=30 es:','\n')
betam
cat('La matriz de covarianzas para n=30 es:','\n')
sigmam
load(file="C:/Simulados/Bernoulli/NoInformativa/EstadisBer1MGMTMI-30.Rdata")
cat('Resumen estadístico para el MGMTMI:','\n')
table30MGMTMI
library(coda)
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGDS1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGDS1-30.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGDS1-30.Rdata")
#Se usan las segundas mitades de las cadenas de cada método para generar
la matriz "cadenas" por método en formato mcm
cadenasMGDS <- mcmc.list(list(mcmc(cadena1MGDS,5001,nrow(cadena1MGDS)),

```

```

                                mcmc(cadena2MGDS,5001,nrow(cadena2MGDS)),
                                mcmc(cadena3MGDS,5001,nrow(cadena3MGDS)))
cat('Resumen estadístico para el MGDS:', '\n')
summary(cadenasMGDS)
#Resultados para el caso 1 cuando beta es (4,3). Bernoulli, n=100
load("C:/Simulados/Bernoulli/NoInformativa/Berbetamsigmam100-1MGTMI.Rdata")
cat('La media de la distribución aproximada para n=100 es:', '\n')
betam
cat('La matriz de covarianzas para n=100 es:', '\n')
sigmam
load(file="C:/Simulados/Bernoulli/NoInformativa/EstadisBer1MGTMI-100.Rdata")
cat('Resumen estadístico para el MGTMI:', '\n')
table100MGTMI
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena1MGDS1-100.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena2MGDS1-100.Rdata")
load(file="C:/Simulados/Bernoulli/NoInformativa/Cadena3MGDS1-100.Rdata")
cadenasMGDS <- mcmc.list(list(mcmc(cadena1MGDS,5001,nrow(cadena1MGDS)),
                                mcmc(cadena2MGDS,5001,nrow(cadena2MGDS)),
                                mcmc(cadena3MGDS,5001,nrow(cadena3MGDS))))
cat('Resumen estadístico para el MGDS:', '\n')
summary(cadenasMGDS)

```