

**ESTIMACIÓN DEL MODELO LOGÍSTICO MIXTO:
REVISIÓN Y NUEVA PROPUESTA**

CLAUDIA PATRICIA GARCÍA VÉLEZ

**MAESTRÍA EN MATEMÁTICAS APLICADAS
ESCUELA DE CIENCIAS Y HUMANIDADES
UNIVERSIDAD EAFIT
MEDELLÍN
2012**

**ESTIMACIÓN DEL MODELO LOGÍSTICO MIXTO:
REVISIÓN Y NUEVA PROPUESTA**

CLAUDIA PATRICIA GARCÍA VÉLEZ

Tesis para optar al título de
Magister en Matemáticas Aplicadas

Director
MYLADIS COGOLLO FLÓREZ, MSc.
Profesora asistente
Departamento de Ciencias Básicas

MAESTRÍA EN MATEMÁTICAS APLICADAS
ESCUELA DE CIENCIAS Y HUMANIDADES
UNIVERSIDAD EAFIT
MEDELLÍN

2012

UNIVERSIDAD EAFIT
ESCUELA DE CIENCIAS Y HUMANIDADES
MAESTRÍA EN MATEMÁTICAS APLICADAS

Los jurados abajo firmantes certifican que han leído y que aprueban la tesis titulada **Estimación de un modelo de regresión logístico mixto: revisión y nueva propuesta** presentada por **Claudia Patricia García Vélez**.

Ciudad y Fecha

Nombre jurado 1

Nombre jurado 1

Nota de aceptación

Jairo Villegas Gutiérrez, MSc.
Coordinador de la Maestría

Myladis Cogollo Flórez, MSc.
Director de la tesis

Ciudad y Fecha

A mi familia y mi mejor amigo.

Agradecimientos

La autora del presente trabajo expresa sus sinceros agradecimientos a:

***Myladis Rocio Cogollo, MSc,** por su asesoría acertada, por su continuo apoyo y su comprensión.*

***Mis padres,** por darme todo de ellos.*

***Juan Pablo M.,** por ser mi mejor amigo y apoyarme en todo.*

Resumen

El proceso de estimación de los parámetros del modelo de regresión logístico mixto, vía máxima verosimilitud, conlleva a expresiones matemáticas que no pueden resolverse analíticamente; por eso, se han propuesto algunos métodos alternativos que permiten obtener una aproximación de las estimaciones empleando integración numérica; entre ellos se destacan el método de máxima verosimilitud simulada y la aproximación Bayesiana usando los algoritmos EM (Esperanza - Maximización) y cuasi-Newton. Sin embargo, estos métodos presentan algunas dificultades computacionales en la integración numérica que se requiere en el proceso debido a la elección de la función logit y al supuesto de normalidad del vector de efectos aleatorios.

En este trabajo se presenta una propuesta basada en técnicas de optimización metaheurísticas, para la estimación de los parámetros del modelo de regresión logística mixto basada en la minimización de la suma de cuadrados de error. Se asume que la distribución del vector de efectos aleatorios es una función de un vector aleatorio con distribución Log-Gamma. Éste último supuesto implica que no se requieran integraciones numéricas para calcular la verosimilitud marginal. Además se considera una función de linkeo alternativa a la función logit, la log-log.

Palabras clave: Modelo Logístico Mixto, Modelo lineal generalizado, Estimación de parámetros.

Contenido

1. Introducción	1
2. Marco teórico	5
2.1. Modelo lineal mixto generalizado	5
2.1.1. Especificación del modelo	6
2.1.2. Modelo logístico mixto	9
2.2. Proceso de estimación de parámetros del modelo logístico mixto	12
2.2.1. Algoritmo EM (Esperanza - Maximización)	12
2.2.2. Bayes Empírico	14
2.2.3. Cuadratura Gaussiana	16
2.2.4. Log-Log	17
2.3. Técnicas Metaheurísticas de Optimización	20
2.3.1. Algoritmos Basados en Cúmulos de Partículas	21
2.3.2. Evolución Diferencial	29
3. Propuesta para la estimación de los parámetros del modelo logístico mixto	32
3.1. Algoritmo de estimación	34
3.2. Prueba de convergencia del algoritmo	37
4. Simulación	43
4.1. Modelo	43
4.2. Selección de parámetros	43
4.3. Resultados	44

4.4. Comparación con un método convencional	57
5. Conclusiones	60
A. Pruebas	67
B. Programas en R	73

Índice de Tablas

2.1. Funciones de enlace inversa asociadas a las funciones de distribución	8
3.1. Características de la función Schwefel 2.26.	38
3.2. Resultados para PSO cuando se realizan 50 corridas para la función (3.2) bidimensional.	39
3.3. Resultados para DE cuando se realizan 50 corridas para la función (3.2) bidimensional.	41
4.1. Resultados numéricos con 1000 iteraciones para $N = 17$ individuos, $CR = 0,7122$ y $F = 0,6301$	46
4.2. Resultados numéricos con 500 iteraciones para $N = 17$ individuos, $CR =$ $0,7122$ y $F = 0,6301$	47
4.3. Resultados numéricos con 100 iteraciones para $N = 17$ individuos, $CR =$ $0,7122$ y $F = 0,6301$	48
4.4. Resultados numéricos con 1000 iteraciones para $N = 47$ individuos, $CR = 0,7122$ y $F = 0,6301$	49
4.5. Resultados numéricos con 500 iteraciones para $N = 47$ individuos, $CR =$ $0,7122$ y $F = 0,6301$	50
4.6. Resultados numéricos con 100 iteraciones para $N = 47$ individuos, $CR =$ $0,7122$ y $F = 0,6301$	51
4.7. Resultados numéricos con 1000 iteraciones para $N = 63$ individuos, $CR = 0,7122$ y $F = 0,6301$	52
4.8. Resultados numéricos con 500 iteraciones para $N = 63$ individuos, $CR =$ $0,7122$ y $F = 0,6301$	53

4.9. Resultados numéricos con 100 iteraciones para $N = 63$ individuos, $CR = 0,7122$ y $F = 0,6301$	54
4.10. Resultados obtenidos con el paquete glmmPQL de R para $N = 50$ individuos, $\beta_0 = -4,8$ y $\beta_1 = 2,3$	58

Índice de Figuras

2.1. Movimiento de una partícula en el espacio de soluciones. Fuente:J.M. García (2006)	24
2.2. Pseudocódigo Algoritmo PSO. Fuente:J.M. García (2006)	26
2.3. Sociometrías de cúmulos <i>globalbest</i> y <i>localbest</i> . Fuente:J.M. García (2006)	27
3.1. Pseudocódigo Algoritmo DE-PSO. Fuente: R. Thangaraj et al. (2011) .	35
3.2. Desempeño del algoritmo PSO Global para optimizar la función Schwefel 2.26.Cada color corresponde al resultado de una corrida para máximo 500 iteraciones, $N = 50$ y $w = 0.7298$	40
3.3. Desempeño del algoritmo PSO Local para optimizar la función Schwefel 2.26.Cada color corresponde al resultado de una corrida para máximo 500 iteraciones, $N = 50$ y $w = 0.7298$	41
3.4. Desempeño del algoritmo DE-PSO para optimizar la función Schwefel 2.26.Cada color corresponde al resultado de una corrida para máximo 400 iteraciones.	42
4.1. Convergencia de una sola corrida	56
4.2. Convergencia de tres corridas. Cada color representa una corrida. . . .	56

Capítulo 1

Introducción

En la actualidad la información estadística proveniente de varios campos de investigación, tales como las ciencias médicas, sociales, económicas, psicológicas y agrícolas, tienen por lo general una estructura jerárquica compleja o de niveles múltiples. Ejemplos de este tipo de estructuras se reconocen en estudios poblacionales donde se recopila la información de los individuos en distintos tiempos, o bien los individuos integran familias, a su vez reunidas en barrios, los cuales conforman ciudades. O también, alumnos agrupados en cursos de diferentes escuelas.

Este agrupamiento, cualquiera sea su origen, provoca observaciones que tienden a estar correlacionadas dentro de los grupos. Es decir, las respuestas de individuos pertenecientes a un mismo grupo tienden a ser más similares que las de diferentes grupos. Ignorar esta relación haciendo uso de las técnicas estadísticas tradicionales puede sobredimensionar los efectos del grupo y por lo tanto distorsionar las conclusiones. En esencia lo que ocurre es que los individuos agrupados proveen menos información que la que hubiesen proporcionado si no tuvieran esa característica en común: pertenecer a algún grupo determinado. Por lo tanto para estudiar si una covariable influye sobre las respuestas de individuos agrupados, según alguna característica en común, el empleo de modelos de regresión clásicos, ignorando el agrupamiento, podría conducir a resultados erróneos.

La modelización de variables en presencia de medidas repetidas es una técnica estadística apropiada para el análisis de este tipo de datos. En términos generales, este enfoque incluye en el modelo efectos aleatorios relacionados con diferentes niveles de información (Goldstein, 2003); lo cual permite manejar dos tipos de fuente de variación: entre grupo e intra grupo.

Si bien la teoría de los modelos de efectos aleatorios con respuesta numérica, usualmente con distribución normal, está bien desarrollada y unificada; no sucede lo mismo cuando la variable respuesta es categórica, para la cual se tienen desarrollos recientes y variados. Éste último tipo de modelos constituye una clase de los denominados modelos lineales mixtos generalizados (GLMM), (Agresti, 2002; Agresti et al, 2000).

En los últimos años, dentro de los modelos GLMM, ha tenido un importante desarrollo en el campo de la investigación y la práctica, el modelo Logístico Mixto. Este modelo es usado para estudiar la relación entre variables explicativas y una respuesta binaria en casos con respuestas dependientes, además se está revelando como el modelo más general y flexible de los disponibles actualmente, pues permite abordar tanto las posibles heteroscedasticidades como las correlaciones entre alternativas.

Aunque muchas aplicaciones del modelo Logístico Mixto aparecen en la literatura (Chintagunta, Jain, and Vilcassim 1991; Gonul and Srinivasan 1993; Jain, Vilcassim, and Chintagunta 1994; and Revelt and Train 1998), pocos sugieren una forma fácil de implementar el procedimiento de estimación de los parámetros del modelo, pues se ha probado su dificultad debido al hecho de que la función de verosimilitud envuelve integrales que no pueden ser derivadas explícitamente (Zeger and Karim, 1991; Breslow and Clayton, 1993; Hedeker and Gibbons, 1996).

Varios métodos han sido propuestos para la estimación de los parámetros del modelo Logístico Mixto, sin embargo no hay ningún método que pueda asumirse que trabaja bien en todas las circunstancias en términos de robustez de los estimadores y eficiencia

computacional. Las técnicas más usadas para obtener estimadores de máxima verosimilitud son el algoritmo EM (Dempster, Lair, and Rubin, 1997), combinación del algoritmo EM con el de Newton (Nash 1990) y el análisis de Bayes Empírico (Lair, and Ware, 1982; Strenio, Weisberg, and Bryk 1983). Tradicionalmente, algunos paquetes estadísticos usan cuadratura Gaussiana para evaluar y maximizar la verosimilitud marginal.

En el presente trabajo no sólo se realiza una revisión sistemática de la literatura de los métodos de estimación de parámetros y sus respectivas desventajas propuestos para el modelo logístico mixto; sino que también se propone una nueva metodología de estimación de parámetros que permite resolver los problemas que se presentan comúnmente en otros métodos de estimación.

La originalidad de la investigación está basada principalmente en tres aspectos:

1. Aunque se han propuesto métodos para la estimación de los parámetros de un modelo Logístico Mixto, no hay consistencia entre ellos, pues al aplicarlos a un mismo conjunto de datos producen estimaciones distintas.
2. En los métodos propuestos no se evalúa la robustez de ellos ante malas especificaciones distribucionales.
3. La consideración de una nueva metodología que mejore el costo computacional y sea robusta ante malas especificaciones distribucionales, permitiría usar este tipo de modelos de manera más rápida y confiable en la modelación de probabilidades de casos de respuesta binaria.

El trabajo esta organizado como sigue. En el capítulo 2 se presenta una introducción general a los modelos lineales mixtos generalizados, haciendo énfasis en el modelo logístico mixto y los diversos métodos de estimación que han sido propuestos hasta la fecha. Además se hace una introducción a las técnicas de optimización metaheurísticas y se profundiza en dos métodos específicos, los algoritmos basados en cúmulos de

partículas (PSO) y los algoritmos de evolución diferencial(DE).

En el capítulo 3 se desarrolla una nueva propuesta de estimación de los parámetros de un modelo con respuesta binaria haciendo uso de la hibridación de las técnicas metaheurísticas descritas en el capítulo anterior y en el capítulo 4 se realiza un estudio de la robustez de la propuesta, vía simulación. Finalmente, en el capítulo 5 se plantean las conclusiones más relevantes del trabajo.

Capítulo 2

Marco teórico

2.1. Modelo lineal mixto generalizado

Los modelos lineales generalizados (GLM), expuestos por McCullagh and Nelder (1989), proveen una extensión de los modelos lineales que relaja los supuestos de normalidad, varianza de error constante y, la relación lineal entre los efectos de las covariables y la media. Un GLM es especificado por:

- i) Una componente aleatoria que determina la distribución de probabilidad de la variable respuesta
- ii) Una componente sistemática, la cual especifica una función lineal de las variables exploratorias usada como predictor
- iii) Una función de enlace que relaciona la componente sistemática y el valor medio de la componente aleatoria.

Por muchos años éstos modelos fueron estandar para el análisis de datos binomiales. La inferencia usando máxima verosimilitud es sencilla y los procesos de diagnóstico fueron desarrollados incluyendo un test para la función de enlace (ver Cheng and Wu; 1994), diagnóstico de residuales (Pierce and Schafer 1986; Cook and Weisberg 1989) y test para el problema de sobre dispersión (Breslow 1989).

Por otra parte, para el caso de medidas repetidas los GLM se extienden a los modelos lineales mixtos generalizados (GLMM), los cuales permiten la flexibilidad asociada a los modelos lineales clásicos con niveles múltiples de efectos aleatorios incorporados en varias categorías. El desarrollo y la utilización de los GLMM fue inicialmente obstaculizado por las intratables integrales que se requieren para la estimación de sus parámetros. Se han propuesto gran variedad de aproximaciones inferenciales, incluyendo cuasi-verosimilitud penalizada, el algoritmo EM, máxima verosimilitud simulada y variaciones de las ecuaciones de estimación generalizadas (GEE).

2.1.1. Especificación del modelo

Suponga que se tiene un esquema de muestreo en dos etapas:

Etapa 1. Seleccione una muestra aleatoria de n individuos de una población de tamaño N . Asocie al i -ésimo individuo, el vector de parámetros específico (efectos aleatorios) $\boldsymbol{\alpha}_i$.

Etapa 2. Condicional a $\boldsymbol{\alpha}_i$, seleccione realizaciones de $\{y_{ij}, x_{ij}\}$, $j = 1, \dots, J_i$, para el i -ésimo individuo.

Suponga que para cada sujeto, $y_{ij}|\boldsymbol{\alpha}_i$ son independientes, tal que

$$f_i(y_{ij}|\boldsymbol{\alpha}_i) = \exp \left\{ \frac{y_{ij}\xi_{ij} - b(\xi_{ij})}{a_i(\phi)} + c_i(y_{ij}, \phi) \right\},$$

donde $b(\cdot)$, $a_i(\cdot)$, $c(\cdot, \cdot)$ son funciones conocidas, y ϕ es un parámetro de dispersión el cual puede ser o no conocido. ξ_{ij} es asociado con $\mu_{ij} = E(y_{ij}|\boldsymbol{\alpha}_i)$, el cual, a su vez, es asociado con un predictor lineal

$$\boldsymbol{\eta}_i = \mathbf{Z}_i\boldsymbol{\alpha}_i + \mathbf{X}_i\boldsymbol{\beta},$$

a través de una función de enlace conocida $g(\cdot)$ tal que $g(\boldsymbol{\mu}_i) = \boldsymbol{\eta}_i$. En este caso, \mathbf{X}_i ($J_i \times p$) y \mathbf{Z}_i ($J_i \times q$) son las matrices diseño asociadas a los efectos fijos y efectos aleatorios del i -ésimo individuo, respectivamente; y $\boldsymbol{\beta}$ es el vector p -dimensional de efectos fijos.

Nótese que en el GLMM intervienen:

- i) Un predictor lineal, η , que se forma como una combinación de los efectos fijos y aleatorios,
- ii) Una función de enlace y/o función de enlace inversa que relaciona la media condicional μ con el predictor lineal η .
- iii) Una función de varianza conocida, a través de la cual la matriz de covarianza $Var[y_{ij}|\boldsymbol{\alpha}_i] = \mathbf{R}$ depende de μ_{ij} .

Por otra parte, un modelo lineal mixto generalizado usa una aproximación para el modelo de variabilidad residual, de tal forma que la relación entre el predictor lineal y el vector de observaciones es modelada como

$$\mathbf{Y}_i|\boldsymbol{\alpha}_i \sim F^*(g(\boldsymbol{\eta}_i), \mathbf{R}),$$

es decir, $\mathbf{Y}_i|\boldsymbol{\alpha}_i$ tiene una distribución F^* con media $g(\boldsymbol{\eta}_i)$ y varianza \mathbf{R} ; donde $g(\cdot)$ es la función de enlace inversa. A F^* se le denota distribución del error.

En las siguientes subsecciones se presenta una definición más clara y descriptiva de la función de enlace inversa y de la función de varianza.

Función de enlace inversa

La función de enlace inversa es usada para mapear el valor del predictor lineal η , en la media condicional μ , para cada observación. La selección de la función de enlace se basa comunmente en la distribución del error y en la búsqueda de simplicidad y habilidad para interpretar los resultados del análisis.

Las funciones de distribución F^* más usadas son: la distribución Normal con la función de enlace Identidad, la distribución Binomial con las funciones de enlace Logit o Probit, la distribución Poisson con la función de enlace Log, y la distribución Gamma con la función enlace Inversa (tabla 2.1).

Tabla 2.1: Funciones de enlace inversa asociadas a las funciones de distribución

Distribución	Función de enlace	Función de enlace inversa	Función de varianza
Normal	Identidad	η	1
Binomial	Logit	$\exp(n)/(1 + \exp(n))$	$\mu(1 - \mu)/n$
	Probit	$\Phi(\eta)$	
Poisson	Log	$\exp(n)$	μ
Gamma	Inversa	$1/n$	μ^2

Para respuestas dicotómicas la función de enlace usada frecuentemente es la Logit: $\eta_{ij} = \ln(\mu_{ij}/(1 - \mu_{ij}))$. La función de enlace Logit, a diferencia de la función de enlace identidad, siempre dará medias estimadas en un rango de cero a uno. Sin embargo, el efecto de cambio de una unidad en el predictor lineal no es constante, es decir, un incremento en el predictor lineal resulta en un incremento de la media condicional, pero no en una tasa constante.

El uso de la función de enlace Logit se ha hecho común por varias razones, entre ellas es de resaltar que ésta función permite tratar los efectos aleatorios del modelo como un parámetro de molestia o de interés secundario para obtener estimadores de máxima verosimilitud condicional de los efectos fijos sin tener que especificar una distribución para los efectos del individuo. Además, la función Logit es simétrica alrededor de $\mu = 1/2$.

El modelo lineal mixto generalizado cuya función de enlace es la función "Logit", se conoce como modelo de regresión logística mixto o modelo Logit con efectos aleatorios y su especificación para el caso de datos binarios se da en la siguiente sección.

Función de varianza

La función de varianza se usa para modelar la variabilidad no sistemática. En el modelo lineal generalizado la variabilidad residual surge de dos fuentes: de la distribución de la muestra y de la sobre dispersión. La variabilidad debida a la distribución de la muestra se modela de forma sencilla, para las distribuciones mencionadas en la tabla(2.1) se tienen las funciones de varianza especificadas en la misma, respectivamente.

La variabilidad debida a la sobre dispersión puede ser modelada de diferentes formas:

- I) Escalar la variabilidad residual como $Var(y_{ij}|\boldsymbol{\mu}) = \phi V(\mu_{ij})$, donde ϕ es el parámetro de sobre dispersión.
- II) Adicionar un efecto aleatorio $e_i \sim N(0, \phi)$ al predictor lineal para cada observación.
- III) Seleccionar otra distribución para las respuestas de los individuos.

Las tres aproximaciones requieren la estimación del parámetro adicional, ϕ . Escalar la variabilidad residual es la aproximación más simple, pero puede llevar a resultados poco satisfactorios. Por otra parte, la adición de un efecto aleatorio trae un gran incremento en el costo computacional.

2.1.2. Modelo logístico mixto

Suponga que se tienen n individuos y para el i -ésimo individuo se tienen J_i observaciones, de tal forma que se tomaron en total $N = \sum_{i=1}^n J_i$ datos. Sea $\mathbf{Y}_i = (y_{i1}, \dots, y_{iJ_i})$ el vector de respuestas del individuo i , con $i = 1, \dots, n$, donde y_{ij} denota la j -ésima respuesta del individuo i . Se asume que y_{ij} es una variable aleatoria dicotómica con distribución Bernoulli de parámetro $p_{ij} = P(y_{ij} = 1|\boldsymbol{\alpha}_i)$, tal que

$$\text{Logit}(p_{ij}) = \eta_i = X_{ij}\boldsymbol{\beta} + Z_{ij}\boldsymbol{\alpha}_i \quad (2.1)$$

donde:

X_{ij} : j -ésima fila de la matriz de diseño para los efectos fijos asociados al individuo i ,

$\mathbf{X}_{i(J_i \times p)}$.

$\boldsymbol{\beta}_{(p \times 1)}$: Vector de efectos fijos.

Z_{ij} : j -ésima fila de la matriz de diseño para los efectos aleatorios asociados al individuo i , $\mathbf{Z}_{i(J_i \times k)}$.

$\boldsymbol{\alpha}_i(k \times 1)$: Vector de efectos aleatorios del individuo i .

A nivel poblacional se asume que las observaciones de un mismo individuo son independientes, así que los efectos aleatorios solo varían entre individuos. Además los $\boldsymbol{\alpha}_i$ se suponen independientes e idénticamente distribuidos de acuerdo a una función de distribución h , esto es, $\boldsymbol{\alpha}_i \sim_{iid} h$.

A partir de (2.1) se obtiene la siguiente especificación para el modelo de regresión logística mixto

$$P(y_{ij} = 1 | \boldsymbol{\alpha}_i) = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}, \quad \text{con } \eta_i = X_{ij}\boldsymbol{\beta} + Z_{ij}\boldsymbol{\alpha}_i \quad (2.2)$$

Esta especificación del modelo incluye tres casos especiales, el modelo de intercepto aleatorio, cuando $Z_{ij} = 1$; el modelo de pendiente aleatoria, cuando $Z_{ij} = X_{ij}$; y el modelo de pendiente e intercepto aleatorio, cuando $Z_{ij} = (1, X_{ij})$.

Para muchos problemas de medidas repetidas, un modelo de intercepto aleatorio es suficiente para modelar la variabilidad intra individual. Además la estructura simple de los efectos aleatorios de este modelo facilita el proceso de estimación, aunque en muchas ocasiones, donde la función de verosimilitud no tiene una forma explícita, la estimación requiere del uso exhaustivo de métodos numéricos tales como cuadratura Gauss-Hermite, aproximación de Laplace, Monte Carlo, entre otros.

Modelo Logístico con intercepto aleatorio

Con el uso de la función Logit, el modelo Logístico de intercepto aleatorio está dado por

$$\text{Logit}(p_{ij}) = \alpha_i + X_{ij}\boldsymbol{\beta} \quad (2.3)$$

lo cual equivale a:

$$P(y_{ij} = 1|\alpha_i) = \frac{\exp(\alpha_i + X_{ij}\boldsymbol{\beta})}{1 + \exp(\alpha_i + X_{ij}\boldsymbol{\beta})} \quad (2.4)$$

donde los $\{\alpha_i\}$ son una muestra aleatoria de una población con densidad $h(\alpha|\boldsymbol{\theta})$ siendo $\boldsymbol{\theta}$ el vector de parámetros de dicha función.

Un análisis condicional para la estimación del modelo es relevante cuando los parámetros fijos, $\boldsymbol{\beta}$, son de interés primario, pues cualquier información acerca de la distribución de los efectos aleatorios se pierde en el condicionamiento. Cuando ambos parámetros son de interés, una alternativa para el análisis condicional esta basada en el supuesto de una distribución específica para los efectos aleatorios.

Como prioridad del modelamiento estadístico, se busca la estimación de los parámetros $\boldsymbol{\beta}$ y de la distribución de α_i , $h(\boldsymbol{\theta})$, vía máxima verosimilitud. Así pues, dado que $y_{ij} \sim \text{Ber}(p_{ij})$, entonces su función de probabilidad está dada por

$$P(y_{ij}; \boldsymbol{\beta}|\alpha_i) = (p_{ij})^{y_{ij}}(1 - p_{ij})^{1-y_{ij}} \quad (2.5)$$

Usando (2.4) se puede expresar (2.5) de esta manera

$$\begin{aligned} P(y_{ij}; \boldsymbol{\beta}|\alpha_i) &= \left[\frac{\exp(\alpha_i + X_{ij}\boldsymbol{\beta})}{1 + \exp(\alpha_i + X_{ij}\boldsymbol{\beta})} \right]^{y_{ij}} \left[1 - \frac{\exp(\alpha_i + X_{ij}\boldsymbol{\beta})}{1 + \exp(\alpha_i + X_{ij}\boldsymbol{\beta})} \right]^{1-y_{ij}} \\ &= \frac{[\exp(\alpha_i + X_{ij}\boldsymbol{\beta})]^{y_{ij}}}{\exp(\alpha_i + X_{ij}\boldsymbol{\beta}) + 1} \end{aligned}$$

Debido a la independencia de las respuestas de cada individuo, la función de probabilidad conjunta aportada por el i -ésimo individuo es

$$P(\mathbf{Y}_i; \boldsymbol{\beta}|\alpha_i) = \prod_{j=1}^{J_i} \frac{[\exp(\alpha_i + X_{ij}\boldsymbol{\beta})]^{y_{ij}}}{\exp(\alpha_i + X_{ij}\boldsymbol{\beta}) + 1}$$

Ahora, aplicando el método de estimación de máxima verosimilitud marginal se obtiene la función de verosimilitud marginal para el i -ésimo individuo

$$L_i(\boldsymbol{\beta}, \boldsymbol{\theta}) = \int P(\mathbf{Y}_i; \boldsymbol{\beta}|\alpha_i)h(\alpha_i, \boldsymbol{\theta})d\alpha_i \quad i = 1, \dots, n$$

Finalmente, la función de verosimilitud cuya maximización permite alcanzar la estimación de β y θ está dada por

$$L(\beta, \theta) = \prod_{i=1}^n L_i(\beta, \theta) = \prod_{i=1}^n \int P(\mathbf{Y}_i; \beta | \alpha_i) h(\alpha_i, \theta) d\alpha_i \quad (2.6)$$

La integral que aparece en la expresión anterior se define como el valor esperado de la probabilidad condicional $P(\mathbf{Y}_i; \beta | \alpha_i)$ tomada sobre todos los posibles valores de la variable no observada α_i y su evaluación numérica es necesaria para continuar el desarrollo analítico de la estimación, pero no tiene solución analítica lo cual representa la mayor dificultad en el proceso de estimación del modelo de regresión logística de intercepto aleatorio.

2.2. Proceso de estimación de parámetros del modelo logístico mixto

Varios autores han propuesto distintos métodos para solucionar la expresión (2.6) en el proceso de estimación de los parámetros del modelo de regresión logístico mixto, basándose en el algoritmo EM, una aproximación empírica de Bayes y el uso de otras funciones de enlace en vez de la función Logit, entre otros.

2.2.1. Algoritmo EM (Esperanza - Maximización)

El algoritmo EM es un procedimiento de optimización para computar estimadores de máxima verosimilitud, generalmente en ajustes de datos incompletos. En el ajuste de datos longitudinales, se consideran los parámetros por individuo, α , como datos omitidos, así que invoca el ajuste de datos incompletos.

Este algoritmo, inicialmente propuesto por Dempster, Laird y Rubin (1977), presenta una técnica iterativa general para realizar una estimación de máxima verosimilitud y puede aplicarse en muchas situaciones. En forma específica para el caso de los modelos

lineales mixtos generalizados, el algoritmo EM basado en unos valores iniciales, repite la siguiente pareja de pasos hasta que satisface un criterio de convergencia impuesto por el investigador:

- 1) **Etapa de estimación (E)**: Calcular la función

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}') = E[\log L'(\boldsymbol{\theta}|\mathbf{Y}_i)]$$

como una función de $\boldsymbol{\theta}$ bajo la suposición de que $\boldsymbol{\theta}'$ es el vector de valores actuales de los parámetros a estimar y donde

$$\log L'(\boldsymbol{\theta}|\mathbf{Y}_i) = \sum_{i=1}^n \log f(\mathbf{Y}_i|\alpha_i) + \log h(\alpha_i|\boldsymbol{\theta})$$

- II) **Etapa de maximización (M)**: Buscar $\boldsymbol{\theta}$ que maximiza la función $Q(\boldsymbol{\theta}|\boldsymbol{\theta}')$ para sustituir $\boldsymbol{\theta}'$.

A pesar de que el algoritmo EM ha sido aplicado con éxito en diversos contextos, puede ser muy lento para converger. En este caso principalmente porque la estimación obtenida depende esencialmente del valor inicial y porque el paso E puede ser analíticamente intratable y se debe acudir a otros métodos de aproximación dentro del algoritmo.

Se han propuesto otros métodos de estimación de parámetros usando el algoritmo EM, sin embargo, a pesar de ser métodos conceptualmente simples, computacionalmente presentan mucha dificultad. Andersen (1980) evitó algunos de los problemas computacionales con el proceso de estimación 2 etapas; él sugirió estimar $\boldsymbol{\beta}$ con el estimador $\hat{\boldsymbol{\beta}}$ de máxima verosimilitud condicional y obtener estimadores de los parámetros de la distribución de α_i , maximizando $L(\hat{\boldsymbol{\beta}}, \theta)$, es decir, sustituyendo $\boldsymbol{\beta}$ por $\hat{\boldsymbol{\beta}}$ en la verosimilitud marginal. Las propiedades de estos estimadores no se conocen, Andersen plantea que para muestras suficientemente grandes las propiedades podrían ser similares a las de los estimadores usuales de máxima verosimilitud; sin embargo esta idea no fue demostrada.

2.2.2. Bayes Empírico

Para implementar la aproximación empírica de Bayes en la inferencia del modelo (2.2), Stiratelli, Laird y Ware (1984) asumen que $\boldsymbol{\alpha}_i \sim MVN(0, \mathbf{D})$, con \mathbf{D} desconocida; $\boldsymbol{\beta}$ tiene una distribución apriori dada por $\boldsymbol{\beta} \sim MVN(0, \boldsymbol{\Gamma})$, tal que $\boldsymbol{\Gamma}^{-1} \rightarrow 0$. Así, en principio se podría estimar $\boldsymbol{\beta}$ y \mathbf{D} maximizando la verosimilitud marginal de los datos, dada en este caso por

$$L(\boldsymbol{\beta}, \mathbf{D}) = \prod_{i=1}^n \int P(\mathbf{Y}_i, \boldsymbol{\beta} | \boldsymbol{\alpha}_i) \exp\left(-\frac{1}{2} \boldsymbol{\alpha}_i^t \mathbf{D}^{-1} \boldsymbol{\alpha}_i\right) |\mathbf{D}|^{-1/2} d\boldsymbol{\alpha}_i \quad (2.7)$$

El parámetro aleatorio de cada individuo sería obtenido como

$$\hat{\boldsymbol{\alpha}}_i = E[\boldsymbol{\alpha}_i | \mathbf{Y}_i, \hat{\boldsymbol{\beta}}, \hat{\mathbf{D}}] \quad (2.8)$$

Esta aproximación tiene dos desventajas:

- 1) El estimador de la varianza es sesgado para muestras pequeñas, el sesgo crece cuando la dimensión de $\boldsymbol{\beta}$ aumenta. Una solución para este problema es usar el estimador de máxima verosimilitud restringida (REML) de la varianza; se estima \mathbf{D} maximizando la función

$$L_R(\mathbf{D}, \boldsymbol{\Gamma}^{-1} = 0) = \lim_{\boldsymbol{\Gamma}^{-1} \rightarrow 0} \left\{ \int L(\boldsymbol{\beta}, \mathbf{D}) \exp\left(-\frac{1}{2} \boldsymbol{\beta}^t \boldsymbol{\Gamma}^{-1} \boldsymbol{\beta}\right) |\boldsymbol{\Gamma}|^{-1/2} d\boldsymbol{\beta} \right\} \quad (2.9)$$

y así el estimador asociado de $\boldsymbol{\beta}$ es

$$\hat{\boldsymbol{\beta}} = E[\boldsymbol{\beta} | \mathbf{Y}, \hat{\mathbf{D}}, \boldsymbol{\Gamma}^{-1} = 0] \quad (2.10)$$

donde $\mathbf{Y}^t = (\mathbf{Y}_1^t, \dots, \mathbf{Y}_n^t)$, la esperanza es sobre la distribución posterior conjunta de $\boldsymbol{\beta}$ y $\boldsymbol{\alpha}^t = (\boldsymbol{\alpha}_1^t, \dots, \boldsymbol{\alpha}_n^t)$.

Las matrices de varianza y covarianza de $\hat{\boldsymbol{\beta}}$ y $\hat{\boldsymbol{\alpha}}$ pueden ser estimadas correspondientemente por

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \text{var}[\boldsymbol{\beta} | \mathbf{Y}, \mathbf{D}, \boldsymbol{\Gamma}^{-1} = 0] \quad (2.11)$$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\alpha}} = \text{var}[\boldsymbol{\alpha} | \mathbf{Y}, \mathbf{D}, \boldsymbol{\Gamma}^{-1} = 0] \quad (2.12)$$

II) La forma cerrada de las expresiones para las integrales en (2.7) y (2.9) (e implícitamente en (2.8),(2.10),(2.11) y (2.12)) no existe. La solución propuesta para estimar β y α es usar la moda posterior en vez de usar la media en (2.8) y en (2.10), es decir, aproximar la distribución posterior conjunta de β y α por una distribución normal multivariada que tiene la misma moda y curvatura en la moda que la distribución posterior verdadera. Denotando la distribución posterior conjunta de β y α como $p(\beta, \alpha | \mathbf{Y}, \mathbf{D}, \Gamma)$, se define $\hat{\beta}$, $\hat{\alpha}$ y $\hat{\Sigma}_{\beta\alpha}$ por

$$p(\hat{\beta}, \hat{\alpha}) = \sup_{\beta, \alpha} p(\beta, \alpha | \mathbf{Y}, \hat{\mathbf{D}}, \Gamma^{-1} = 0) \quad (2.13)$$

y

$$\hat{\Sigma}_{\beta\alpha} = - \left\{ \frac{\partial^2 \ln p(\hat{\beta}, \hat{\alpha} | \mathbf{Y}, \hat{\mathbf{D}}, \Gamma^{-1} = 0)}{\partial(\beta, \alpha) \partial(\beta, \alpha)^t} \right\}^{-1} \quad (2.14)$$

donde $\hat{\mathbf{D}}$ maximiza (2.9). Como $p(\beta, \alpha | \mathbf{Y}, \hat{\mathbf{D}}, \Gamma^{-1} = 0)$ es proporcional al producto de $f(\mathbf{Y} | \beta, \alpha)$ y $h(\alpha | \hat{\mathbf{D}})$, entonces $\hat{\beta}$ y $\hat{\alpha}$ pueden calcularse usando el procedimiento estandar similar al usado en máxima verosimilitud ordinaria. Además $\hat{\Sigma}_{\beta\alpha}^{-1}$ es realizable.

Para estimar \mathbf{D} se han presentado distintas aproximaciones, una de ellas propuesta por Laird (1977) consiste en tomar la maximización de (2.9) como objetivo y aproximar sus derivadas calculando $\hat{\mathbf{D}}$, esta aproximación se hace usando el algoritmo EM así

Etapas M: Suponer que α fue observado. Entonces el MLE de \mathbf{D} podría ser

$$\hat{\mathbf{D}} = \sum_{i=1}^n \alpha_i \alpha_i^t / n \quad (2.15)$$

Etapas E: Calcular el valor esperado del estadístico eficiente

$$\hat{t} = \sum_{i=1}^n E[\alpha_i \alpha_i^t | \mathbf{Y}_i, \hat{\mathbf{D}}, \Gamma^{-1} = 0] \quad (2.16)$$

El calculo exacto de (2.16) requiere n integraciones numéricas. Sin embargo si se usa la aproximación normal que se asumió anteriormente, el calculo se vuelve

sencillo. Es decir, se asume que la distribución condicional de α , dado Y , es aproximadamente normal con media $\hat{\alpha}$ y matriz de covarianza $\hat{\Sigma}_{\alpha}$ como se define en (2.13) y (2.14). El calculo de $\hat{\alpha}$ generalmente requiere iteración, así que se usa un proceso iterativo como el de Newton-Raphson en cada etapa E del algoritmo EM.

Este método puede ser muy costoso en el análisis exploratorio para algunos casos, en otros puede llegar a ser más eficiente usar la integración numérica para aproximar (2.16) en la etapa E.

2.2.3. Cuadratura Gaussiana

Otro método usado para lograr la estimación de los parámetros del modelo Logit y en general de los modelos lineales mixtos generalizados, es la cuadratura Gaussiana. La cuadratura puede ser útil en la aproximación de la verosimilitud marginal por integración numérica. Específicamente, si la distribución asumida para los efectos aleatorios es normal, la cuadratura Gauss-Hermite puede aproximar la integral a un grado alto de exactitud. La integral es aproximada por una sumatoria sobre un número especificado de puntos de cuadratura Q para cada dimensión de la integración. Si se asume que los efectos aleatorios se rigen por una distribución distinta de la normal, se escogen otros puntos de cuadratura pero la exactitud es más débil.

En el método la función de probabilidad $f(\mathbf{Y}_i|\boldsymbol{\theta})$ es obtenida sustituyendo los parámetros $\boldsymbol{\theta}$ por el vector r -dimensional de puntos de cuadratura Q , así la verosimilitud marginal es aproximada como:

$$L(\boldsymbol{\theta}) \approx \sum_{q=1}^r \prod_{i=1}^n f(\mathbf{Y}_i|Q_r)A(Q_r)$$

donde $A(Q_r)$ es la evaluación de los puntos de cuadratura en la distribución de los efectos aleatorios, es decir $h(\alpha_i|Q_r) = A(Q_r)$.

La cuadratura Gaussiana puede funcionar bien si las respuestas son dicotómicas y el tamaño de los cluster o grupos de datos, es pequeño. Sin embargo, frecuentemente se

necesita un gran número de puntos de cuadratura para obtener una buena aproximación de la verosimilitud. Como consecuencia, el método puede ser computacionalmente intensivo, particularmente si hay muchos efectos aleatorios. Es importante anotar que la cuadratura Gaussiana puede fallar aún para datos binarios y grupos pequeños si la correlación intraindividual es muy alta.

2.2.4. Log-Log

Un método alternativo, basado en la maximización de la verosimilitud marginal para analizar datos binarios con efectos aleatorios fue propuesto por Conaway (1990). Muestra que asumiendo una familia paramétrica que sugiera una amplia variedad de formas para la distribución de los efectos aleatorios, la verosimilitud marginal puede ser computada sin integraciones numéricas.

La propuesta de Conaway se enfoca en el uso de la función de enlace "Log-Log", $g(p_{ij}) = \ln(-\ln(p_{ij}))$, como una alternativa a la función Logit. La distribución asumida para los efectos aleatorios es la distribución Log-gamma, la cual provee una amplia variedad de formas para la distribución de los efectos aleatorios, incluyendo algunas que parecen cercanas a la distribución normal. Así, no es difícil maximizar la verosimilitud marginal resultante, pues no se requiere integración numérica para su evaluación.

El modelo planteado por Conaway (1990) es el siguiente:

$$\ln(-\ln(p_{ij})) = \alpha_i + \mathbf{X}_{ij}\boldsymbol{\beta}$$

el cual se puede reescribir, usando el enlace inverso, como

$$p_{ij} = \exp(-\exp(\alpha_i + \mathbf{X}_{ij}\boldsymbol{\beta})) \tag{2.17}$$

Asumir la distribución Log-gamma para los α_i es equivalente a asumir la distribución Gamma para la transformación $\exp(\alpha_i)$ (ver demostración en A.1). Por lo tanto, si los α_i son independientes e idénticamente distribuidos con $\exp(\alpha_i) \sim \text{gamma}(k, \lambda)$, la función

de densidad de α_i , es dada por

$$h(\alpha_i|k, \lambda) = \frac{1}{\lambda^k \Gamma(k)} \exp(k\alpha_i) \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right) \quad (2.18)$$

En el teorema (A.1) se demuestra que bajo la función de densidad anterior se pueden determinar las siguientes expresiones para la media y la varianza de α_i , tal como lo plantea Conaway

$$E(\alpha_i) = \psi(k) + \ln(\lambda)$$

y

$$Var(\alpha_i) = \psi'(k)$$

donde $\Gamma(k)$ es la función gamma, $\Gamma(k) = \int t^{k-1} \exp^{-t} dt$; $\psi(k)$ es la derivada del logaritmo de la función gamma y $\psi'(k)$ es la derivada de $\psi(\cdot)$.

Además de los supuestos anteriores, el asumir independencia entre las respuestas del i -ésimo individuo permite escribir la probabilidad conjunta de las respuestas del individuo así

$$P(Y_i|\alpha_i) = \prod_{j=1}^{J_i} P(y_{ij} = 1|\alpha_i) \quad (2.19)$$

Combinando los supuestos (2.17),(2.18) y (2.19), se puede determinar una expresión analítica de la esperanza condicional de las respuestas, variando sobre la distribución de los efectos aleatorios de esta manera

$$\begin{aligned} P(y_{ij} = 1) &= E(y_{ij}) = E[E(y_{ij}|\alpha_i)] \\ &= \int P(y_{ij} = 1|\alpha_i) h(\alpha_i|k, \lambda) d\alpha_i \\ &= \left(\frac{1/\lambda}{1/\lambda + \exp(X_{ij}\boldsymbol{\beta})} \right)^k \end{aligned} \quad (2.20)$$

Análogo al proceso que permite encontrar la expresión anterior, los supuestos permiten determinar una solución analítica de la integral que surge en la función de

verosimilitud marginal del i -ésimo individuo, sin necesidad de utilizar integración numérica. Así que en este caso la función de verosimilitud marginal está dada por

$$\begin{aligned} L_i(\boldsymbol{\beta}, k, \lambda) &= \int \prod_{j=1}^{J_i} P(y_{ij} = 1 | \alpha_i) h(\alpha_i | k, \lambda) d\alpha_i \\ &= \left(\frac{1/\lambda}{1/\lambda + \sum_{j=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta})} \right)^k \end{aligned} \quad (2.21)$$

Las demostraciones (A.2) y (A.3) muestran el desarrollo que lleva a las igualdades en (2.20) y en (2.21) repectivamente.

De (2.21) se llega a la función log-verosimilitud marginal a partir de la cual Conaway sugiere obtener los estimadores de los parámetros a través de un método de maximización que consiste en obtener el vector de probabilidades de todos los individuos resolviendo un sistema de ecuaciones que involucra la matriz de diseño y un vector con las probabilidades de los individuos, pero calculadas sobre todos los posibles subconjuntos de un subconjunto de respuestas de los individuos.

Por otra parte, si se desea emplear el método de máxima verosimilitud tradicional se obtiene que

$$\begin{aligned} \ln L(\boldsymbol{\beta}, k, \lambda) &= \sum_{i=1}^n \ln L_i(\boldsymbol{\beta}, k, \lambda) = \sum_{i=1}^n k \ln \left(\frac{1/\lambda}{1/\lambda + \sum_{j=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta})} \right) \\ &= \sum_{i=1}^n k \left[\ln \frac{1}{\lambda} - \ln \left(\frac{1}{\lambda} + \sum_{j=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta}) \right) \right] \\ &= -k \sum_{i=1}^n \ln \left(1 + \lambda \sum_{j=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta}) \right) \end{aligned}$$

Y la optimización del problema, es decir los estimadores, se alcanzan hallando las raíces de las siguientes ecuaciones

$$\begin{aligned}
\frac{\partial}{\partial k} \ln[L(\boldsymbol{\beta}, k, \lambda)] &= - \sum_{i=1}^n \ln \left(1 + \lambda \sum_{j=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta}) \right) \\
\frac{\partial}{\partial \lambda} \ln[L(\boldsymbol{\beta}, k, \lambda)] &= -k \sum_{i=1}^n \left(\frac{\sum_{j=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta})}{1 + \lambda \sum_{j=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta})} \right) \\
\frac{\partial}{\partial \beta_t} \ln[L(\boldsymbol{\beta}, k, \lambda)] &= -k \sum_{i=1}^n \left(\frac{\lambda \sum_{j=1}^{J_i} X_{ijt} \exp(X_{ij}\boldsymbol{\beta})}{1 + \lambda \sum_{j=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta})} \right), \quad t = 1, 2, \dots, p
\end{aligned}$$

Pero este último sistema de ecuaciones no tiene solución, por lo tanto el proceso de estimación no puede completarse con el método usual de maximización.

2.3. Técnicas Metaheurísticas de Optimización

Las técnicas metaheurísticas son cada vez más usadas en la resolución de problemas de optimización conforme estos se hacen más grandes y complejos. Estas técnicas, tal como las exponen Glover y Kochenberger en su libro *Handbook of Metaheuristics*, son estrategias de alto nivel que combinan diferentes métodos heurísticos para explorar el espacio de búsqueda de soluciones del problema de forma eficiente y efectiva.

Hay diferentes formas de clasificar y describir las técnicas metaheurísticas, dependiendo de las características que se seleccionen; una de las clasificaciones más utilizada divide las metaheurísticas en dos: basadas en trayectorias y basadas en población.

Las técnicas basadas en trayectorias se caracterizan por la búsqueda de la solución óptima partiendo de un punto y mediante la exploración del vecindario de éste, se va actualizando la solución actual, formando una trayectoria; normalmente se termina la búsqueda cuando se alcanza un número máximo de iteraciones predefinido, se encuentra una solución con una calidad aceptable o se detecta un estancamiento del proceso.

Las técnicas basadas en la población se caracterizan por trabajar con un conjunto de soluciones en cada iteración, el resultado final de este tipo de algoritmos depende

fuertemente de la forma en que se manipule la población. Una de las metaheurísticas que hace parte de esta clasificación es el Algoritmo Basado en Cúmulos de Partículas o Particle Swarm Optimization (PSO), que se describirá en la siguiente sección.

2.3.1. Algoritmos Basados en Cúmulos de Partículas

Como se mencionó antes, un Algoritmo Basado en Cúmulos de Partículas es una técnica metaheurística basada en poblaciones. El PSO fue originalmente desarrollado por el psicólogo-sociólogo James Kennedy y por el ingeniero electrónico Russell Eberhart en 1995, basándose en un enfoque conocido como la "metáfora social" que describe a este algoritmo y que se puede resumir de la siguiente forma: los individuos que conviven en una sociedad tienen una opinión que es parte de un *conjunto de creencias* (el espacio de búsqueda) compartido por todos los posibles individuos. Cada individuo puede modificar su propia opinión basándose en tres factores:

- Su conocimiento sobre el entorno (su valor de fitness).
- Su conocimiento histórico o experiencias anteriores (su memoria).
- El conocimiento histórico o experiencias anteriores de los individuos situados en su vecindario.

Siguiendo ciertas reglas de interacción, los individuos en la población adaptan sus esquemas de creencias al de los individuos con más éxito de su entorno. Con el tiempo, surge una cultura cuyos individuos tienen un conjunto de creencias estrechamente relacionado.

El principio natural en el que se basa PSO es el comportamiento de una bandada de aves o de un banco de peces: supongamos que una de estas bandadas busca comida en un área y que solamente hay una pieza de comida en dicha área. Los pájaros no saben dónde está la comida pero sí conocen su distancia a la misma, por lo que la estrategia más eficaz para hallar la comida es seguir al ave que se encuentre más cerca de ella. PSO emula este escenario para resolver problemas de optimización. Cada solución (partícula) es un *ave* en el espacio de búsqueda que está siempre en continuo movimiento y que

nunca muere.

El cúmulo de partículas es un sistema multiagente, es decir, las partículas son agentes simples que se mueven por el espacio de búsqueda y que guardan (y posiblemente comunican) la mejor solución que han encontrado. Cada partícula tiene un fitness o función de bondad que cuantifica su grado de adecuación, una posición y un vector velocidad que dirige su "movimiento"; este movimiento de las partículas por el espacio está guiado por las partículas óptimas en el momento actual. Para cada partícula se define el vecindario como el conjunto de soluciones que se pueden construir a partir de ella aplicando un movimiento.

Descripción del Algoritmo PSO

Un algoritmo PSO consiste en un proceso iterativo y estocástico que opera sobre un cúmulo de partículas. La posición de cada partícula representa una solución potencial al problema que se está resolviendo. Generalmente, una partícula p_i está compuesta de tres vectores y dos valores de fitness:

- El vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})$ almacena la posición actual (localización) de la partícula en el espacio de búsqueda.
- El vector $\mathbf{pBest}_i = (p_{i1}, p_{i2}, \dots, p_{in})$ almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector de velocidad $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{in})$ almacena el gradiente (dirección) según el cual se moverá la partícula.
- El valor de fitness $fitness_{x_i}$ almacena el valor de adecuación de la solución actual (vector \mathbf{x}_i).
- El valor de fitness $fitness_{pBest_i}$ almacena el valor de adecuación de la mejor solución local encontrada hasta el momento (vector \mathbf{pBest}_i).

El cúmulo se inicializa generando las posiciones y las velocidades iniciales de las partículas. Las posiciones se pueden generar aleatoriamente en el espacio de búsqueda

(puede ser con ayuda de un heurístico de construcción), de forma regular o con una combinación de ambas formas. Una vez generadas las posiciones, se calcula el fitness de cada una y se actualizan los valores de $fitness_{x_i}$ y $fitness_{pBest_i}$.

Las velocidades se generan aleatoriamente, con cada componente en el intervalo $[-v_{max}, v_{max}]$, donde v_{max} será la velocidad máxima que pueda tomar una partícula en cada movimiento. No es conveniente fijarlas a cero pues no se obtienen buenos resultados.

Inicializado el cúmulo, las partículas se deben mover dentro del proceso iterativo. Una partícula se mueve desde una posición del espacio de búsqueda hasta otra, simplemente, añadiendo al vector posición \mathbf{x}_i el vector velocidad \mathbf{v}_i para obtener un nuevo vector posición:

$$\mathbf{x}_i \longleftarrow \mathbf{x}_i + \mathbf{v}_i \quad (2.22)$$

Una vez calculada la nueva posición de la partícula, se evalúa actualizando $fitness_{x_i}$. Además, si el nuevo fitness es el mejor fitness encontrado hasta el momento, se actualizan los valores de mejor posición $pBest_i$ y fitness $fitness_{pBest_i}$. El vector velocidad de cada partícula es modificado en cada iteración utilizando la velocidad anterior, un componente cognitivo y un componente social. El modelo matemático resultante y que representa el corazón del algoritmo PSO viene representado por las siguientes ecuaciones:

$$v_i^{k+1} = \omega v_i^k + \varphi_1 rand_1(pBest_i - x_i^k) + \varphi_2 rand_2(g_i - x_i^k) \quad (2.23)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2.24)$$

La Ecuación 2.23 refleja la actualización del vector velocidad de cada partícula i en cada iteración k . El componente cognitivo está modelado por el factor $\varphi_1 rand_1(pBest_i - x_i^k)$ y representa la distancia entre la posición actual y la mejor conocida por esa partícula, es decir, la decisión que tomará la partícula influenciada por su propia experiencia a

lo largo de su vida. El componente social está modelado por $\varphi_2 rand_2(g_i - x_i^k)$ y representa la distancia entre la posición actual y la mejor posición del vecindario, es decir, la decisión que tomará la partícula según la influencia que el resto del cúmulo ejerce sobre ella. Una descripción más detallada de cada factor se realiza a continuación:

v_i^k : velocidad de la partícula i en la iteración k ,

ω : factor inercia mediante el que se controla el balance entre explotación y exploración en la búsqueda,

φ_1, φ_2 : son tasas de aprendizaje (pesos) que controlan los componentes cognitivo y social, respectivamente

$rand_1, rand_2$: números aleatorios uniformes entre 0 y 1,

x_i^k : posición actual de la partícula i en la iteración k ,

$pBest_i$: mejor posición (solución) encontrada por la partícula i hasta el momento,

g_i : representa la posición de la partícula con el mejor $pBest_fitness$ del entorno de p_i (localbest) o de todo el cúmulo (globalbest).

La ecuación (2.24) modela el movimiento de cada partícula i en cada iteración k . En la figura 2.1 se muestra una representación gráfica del movimiento de una partícula en el espacio de soluciones.

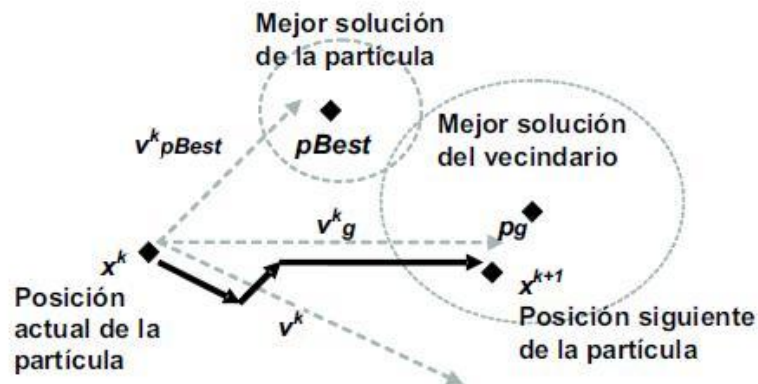


Figura 2.1: Movimiento de una partícula en el espacio de soluciones. Fuente: J.M. García (2006)

En esta gráfica, las flechas de línea discontinua representan la dirección de los vectores de velocidad actual: v_{pBest}^k es la velocidad de la mejor posición tomada por la partícula, v_g^k es la velocidad de la mejor partícula encontrada en el vecindario y v^k es la velocidad actual de la partícula. La flecha de línea continua representa la dirección que toma la partícula para moverse desde la posición x^k hasta la posición x^{k+1} . El cambio de dirección de esta flecha depende de la influencia de las demás direcciones (gradiente) que intervienen en el movimiento.

Se pueden obtener diferentes tipos de PSO atendiendo a diversos tipos de configuración, por ejemplo la importancia de los pesos cognitivo y social y según el tipo de vecindario utilizado.

Por una parte, dependiendo de la influencia de los factores cognitivo y social (valores φ_1 y φ_2 respectivamente) sobre la dirección de la velocidad que toma una partícula en el movimiento (Ecuación 2.24), Kennedy (1997) identifica cuatro tipos de algoritmos:

- Modelo Completo: $\varphi_1, \varphi_2 > 0$. Tanto el componente cognitivo como el social intervienen en el movimiento.
- Modelo sólo Cognitivo: $\varphi_1 > 0$ y $\varphi_2 = 0$. Únicamente el componente cognitivo interviene en el movimiento.
- Modelo sólo Social: $\varphi_1 = 0$ y $\varphi_2 > 0$. Únicamente el componente social interviene en el movimiento.
- Modelo sólo Social exclusivo: $\varphi_1 = 0$ y $\varphi_2 > 0$ y $g_i \neq x_i$. La posición de la partícula en sí no puede ser la mejor de su entorno.

Por otra parte, desde el punto de vista del vecindario, es decir, la cantidad y posición de las partículas que intervienen en el cálculo de la distancia en la componente social, se clasifican dos tipos de algoritmos: PSO Local y PSO Global.

En el PSO Local, se calcula la distancia entre la posición actual de partícula y la posición de la mejor partícula encontrada en el entorno local de la primera. El entorno local consiste en las partículas inmediatamente cercanas en la topología del cúmulo.

Para el PSO Global, la distancia en el componente social viene dada por la diferencia entre la posición de la partícula actual y la posición de la mejor partícula encontrada en el cúmulo completo \mathbf{gBest}_i , como muestra el pseudocódigo en el Algoritmo.

Algoritmo PSO Global

```

S ← InicializarCumulo()
while no se alcance la condición de parada do
  for i = 1 to size(S) do
    evaluar cada partícula  $x_i$  del cúmulo S
    if fitness( $x_i$ ) es mejor que fitness(pBesti) then
      pBesti ←  $x_i$ ; fitness(pBesti) ← fitness( $x_i$ )
    end if
    if fitness(pBesti) es mejor que fitness(gBest) then
      gBest ← pBesti; fitness(gBest) ← fitness(pBesti)
    end if
  end for
  for i = 1 to size(S) do
     $v_i$  ←  $\omega \cdot v_i + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i) + \varphi_2 \cdot rand_2 \cdot (gBest - x_i)$ 
     $x_i$  ←  $x_i + v_i$ 
  end for
end while
Salida: la mejor solución encontrada

```

Figura 2.2: Pseudocódigo Algoritmo PSO. Fuente: J.M. García (2006)

La versión Global converge más rápido pues la visibilidad de cada partícula es mejor y se acercan más a la mejor del cúmulo favoreciendo la intensificación, por esta razón, es la versión que usaremos en la propuesta de estimación del presente trabajo.

El desarrollo de una partícula depende tanto de la topología del cúmulo como de la versión del algoritmo. Las topologías definen el entorno de interacción de una partícula individual con su vecindario. La propia partícula siempre pertenece a su entorno. Los entornos pueden ser de dos tipos:

- *Geográficos*: se calcula la distancia de la partícula actual al resto y se toman las más cercanas para componer su entorno.
- *Sociales*: se define a priori una lista de vecinas para partícula, independientemente de su posición en el espacio.

Los entornos sociales son los más empleados. Una vez definido un entorno, es necesario definir su tamaño, son habituales valores de 3 y 5 pues suelen tener un buen comportamiento. Cuando el tamaño es todo el cúmulo de partículas, el entorno es a la vez geográfico y social, obteniendo así un PSO Global.

Es posible configurar varios tipos de topologías dentro de un entorno social de cúmulo. Una de las más comunes y que inicialmente fue más utilizada es la sociometría *globalbest*; en ésta, se establece un vecindario global en el que toda partícula es vecina de la totalidad del cúmulo (PSO Global) favoreciendo la explotación de espacio de soluciones. Otro ejemplo es la sociometría *localbest* (Figura 2.3), que fue propuesta para tratar con problemas de mayor dificultad. En *localbest* (o anillo), cada partícula es conectada con sus vecinas inmediatas en el cúmulo, así, por ejemplo, la partícula p_i es vecina de la partícula p_{i-1} y de p_{i+1} . Esta topología ofrece la ventaja de establecer subcúmulos que realizan búsquedas en diversas regiones del espacio del problema y de esta forma se favorece la exploración.

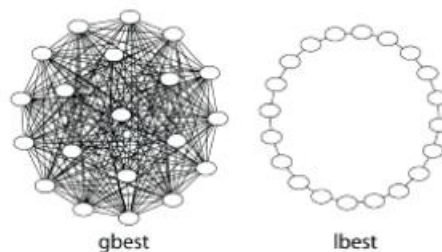


Figura 2.3: Sociometrías de cúmulos *globalbest* y *localbest*. Fuente:J.M. García (2006)

Un problema habitual de los algoritmos de PSO es que la magnitud de la velocidad suele llegar a ser muy grande durante la ejecución, con lo que las partículas se mueven demasiado rápido por el espacio. El rendimiento puede disminuir si no se fija adecuadamente el valor de v_{max} , la velocidad máxima de cada componente del vector velocidad. En la literatura, se habla de dos métodos para controlar el excesivo crecimiento de las velocidades: Un factor de inercia, ajustado dinámicamente y un coeficiente de restricción.

El factor inercia ω , es multiplicado por la velocidad actual en la ecuación de actualización de la velocidad (Ecuación 2.23). ω se va reduciendo gradualmente a lo largo del tiempo (medido en iteraciones del algoritmo). Podemos realizar el cálculo de la inercia utilizando la siguiente ecuación en cada iteración del algoritmo.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} iter$$

Donde ω_{max} es el peso inicial y ω_{min} el peso final, $iter_{max}$ es el número máximo de iteraciones y $iter$ es la iteración actual. ω debe mantenerse entre 0.9 y 1.2. Valores altos provocan una búsqueda exhaustiva (más diversificación) y valores bajos una búsqueda más localizada (más intensificación).

Por otra parte, el coeficiente de restricción introduce una nueva ecuación para la actualización de la velocidad y asegura la convergencia.

$$v_i^{k+1} = K[v_i^k + \varphi_1 rand_1(pBest_i - x_i^k) + \varphi_2 rand_2(g_i - x_i^k)]$$

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \text{ donde } \varphi = \varphi_1 + \varphi_2; \varphi > 4$$

Otro aspecto a tener en consideración es el tamaño del cúmulo de partículas, pues determina el equilibrio entre la calidad de las soluciones obtenidas y el costo computacional (número de evaluaciones necesarias). Recientemente, se han propuesto algunas variantes que adaptan heurísticamente el tamaño del cúmulo, de manera que, si la calidad del entorno de la partícula ha mejorado pero la partícula es la peor de su entorno, se elimina la partícula. Por otra parte, si la partícula es la mejor de su entorno pero no

hay mejora en el mismo, se crea una nueva partícula a partir de ella. Las decisiones se toman de forma probabilística en función del tamaño actual del cúmulo.

2.3.2. Evolución Diferencial

Otros métodos basados en población son los Algoritmos Evolutivos, estos están inspirados en la capacidad de la naturaleza para evolucionar seres para adaptarlos a los cambios de su entorno. Esta familia de técnicas siguen un proceso iterativo y estocástico que opera sobre una población de individuos. Cada individuo representa una solución potencial al problema que se está resolviendo. Inicialmente, la población es generada aleatoriamente (quizás con ayuda de un heurístico de construcción). Cada individuo en la población tiene asignado, por medio de una función de aptitud (fitness), una medida de su bondad con respecto al problema bajo consideración. Este valor es la información cuantitativa que el algoritmo usa para guiar su búsqueda. En los métodos que siguen el esquema de los algoritmos evolutivos, la modificación de la población se lleva a cabo mediante tres operadores: selección, recombinación y mutación. Estos algoritmos establecen un equilibrio entre la explotación de buenas soluciones (fase de selección) y la exploración de nuevas zonas del espacio de búsqueda (fase de reproducción), basados sobre el hecho que la política de reemplazo permite la aceptación de nuevas soluciones que no mejoran necesariamente la existentes.

La Evolución Diferencial (DE) hace parte de los métodos mencionados en el párrafo anterior y particularmente ofrece una forma de optimizar un problema sin usar su gradiente, lo que se hace muy útil en aquellos casos en que el gradiente es difícil o imposible de hallar.

El algoritmo de Evolución Diferencial (DE) fue propuesto por Storn y Price en 1998. Se trata de una técnica no determinista basada en la evolución de una población de vectores (individuos) de valores reales que representan las soluciones en el espacio de búsqueda. La generación de nuevos individuos se lleva a cabo mediante operadores diferenciales de mutación y cruce.

Mediante la mutación diferencial se añade la diferencia proporcional de dos individuos elegidos aleatoriamente de la población a un tercer individuo (también elegido aleatoriamente). Formalmente, dados tres individuos $\mathbf{v}_{r1}, \mathbf{v}_{r2}$ y \mathbf{v}_{r3} elegidos aleatoriamente de la población, donde $r1, r2, r3 \in \{1, 2, \dots, N\}$ son números aleatorios diferentes entre sí y N es el tamaño de la población, un nuevo individuo mutado \mathbf{w}_i se genera mediante la siguiente expresión:

$$\mathbf{w}_i \leftarrow \mathbf{v}_{r1} + \mu(\mathbf{v}_{r2} - \mathbf{v}_{r3}) \quad (2.25)$$

La constante de mutación $\mu > 0$ establece el rango diferenciación entre los individuos \mathbf{v}_{r2} y \mathbf{v}_{r3} con el objetivo de evitar el estancamiento en el proceso de búsqueda.

Tras la mutación, se realiza una operación de recombinación sobre cada individuo \mathbf{v}_i (target) para generar un individuo intermedio \mathbf{u}_i (trial). Esta operación de cruce selecciona uniformemente una posición j del vector trial con la misma probabilidad que del vector target obtenido del individuo mutado.

$$u_i(j) = \begin{cases} w_i(j) & \text{si } r \leq C_r \text{ o } j = j_r \\ v_i(j) & \text{en otro caso} \end{cases}$$

Como se puede observar en la ecuación anterior, el operador de cruce elige aleatoriamente un valor entero $j_r \in [1 \dots D]$ y un valor real aleatorio $r \in (0, 1)$, también uniformemente distribuido para cada componente $j \in (1 \dots D)$ del vector trial \mathbf{u}_i . De este modo, con una probabilidad de recombinación C_r o bien en el caso en el que se cumpla la igualdad $j = j_r$, se selecciona el elemento j_{esimo} del individuo mutado $w_i(j)$ para ser colocado en el elemento j_{esimo} del individuo trial $u_i(j)$. En otro caso, se selecciona el elemento j_{esimo} del individuo target $v_i(j)$ para ser colocado en el elemento j_{esimo} del individuo trial. Finalmente, mediante un operador de selección se decide la aceptación del individuo trial para la nueva generación si consigue alguna mejora sobre el individuo anterior, como muestra la siguiente ecuación

$$\mathbf{v}_i = \begin{cases} \mathbf{u}_i & \text{si } f(\mathbf{u}_i) \leq f(\mathbf{v}_i) \\ \mathbf{v}_i & \text{en otro caso} \end{cases}$$

En la siguiente sección, se describe el algoritmo objeto de este trabajo en el que se utilizan las estructuras y operadores, tanto de PSO como de DE.

Capítulo 3

Propuesta para la estimación de los parámetros del modelo logístico mixto

La propuesta de trabajo, después de estudiar el estado del arte de la estimación de los parámetros del modelo de regresión Logística Mixto y de ver las principales debilidades de los procesos, se basa en la implementación de métodos de estimación usando mecanismos más parsimoniosos y completos que permitan evadir algunos de los problemas que comúnmente se presentan en los métodos estudiados.

En esta investigación se propone un mecanismo de estimación basado en un enfoque de técnicas de optimización metaheurísticas. El objetivo de este desarrollo, a parte de estudiar las ventajas y desventajas del nuevo enfoque, es abrir una ventana al uso de las técnicas metaheurísticas para solucionar problemas de estimación estadística que son difíciles de abordar con las técnicas convencionales.

Teniendo en cuenta que el objetivo principal del estudio del modelo de regresión Logística Mixto es la estimación de la probabilidad para una variable dicotómica, para la presente propuesta se parte del modelo expuesto por Conaway (1990).

Sea $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iJ_i})$ el vector de J_i respuestas del i -ésimo individuo, con $i =$

$1, 2, \dots, n$. En donde, cada y_{ij} es una respuesta dicotómica, tal que $y_{ij} \sim Ber(p_{ij})$ con $p_{ij} = P(y_{ij} = 1)$.

Para modelar la probabilidad p_{ij} teniendo en cuenta la inclusión de un efecto aleatorio que caracterice la correlación entre las observaciones de un mismo individuo, se utiliza la función de enlace Log-Log tal como en (2.17)

$$p_{ij} = P(y_{ij} = 1) = \exp(-\exp(\alpha_i + X_{ij}\boldsymbol{\beta}))$$

donde se asume que los efectos aleatorios, α_i , son variables independientes e idénticamente distribuidas Log-gamma de parámetros k, λ ; $h(\alpha_i|k, \lambda)$ como en (2.18).

A partir de ésta especificación se puede llegar a la siguiente expresión analítica de la esperanza marginal de la respuesta de cada individuo (ver teorema A.2)

$$E[y_{ij}] = \left(\frac{1/\lambda}{1/\lambda + \exp(X_{ij}\boldsymbol{\beta})} \right)^k, \quad (3.1)$$

y con el supuesto de independencia de las respuestas de cada individuo, encontramos una expresión similar para la función de máxima verosimilitud marginal del i -ésimo individuo (ver teorema A.3), dada por

$$L(\boldsymbol{\beta}, k, \lambda) = \left(\frac{1/\lambda}{1/\lambda + \sum_{i=1}^{J_i} \exp(X_{ij}\boldsymbol{\beta})} \right)^k$$

Partiendo de la ecuación anterior, se pueden presentar problemas de invertibilidad y diferenciabilidad a la hora de aplicar métodos numéricos tradicionales en la estimación. Para solucionar estos problemas y obtener una ganancia computacional, en este trabajo se plantea el uso de un algoritmo híbrido Evolución Diferencial-Cúmulo de Partículas, el cual se denota en el resto del documento como DE-PSO. El objetivo principal de este método es aprovechar los puntos fuertes de los algoritmos con el fin de mantener un equilibrio entre los factores de exploración y explotación en el espacio de búsqueda, previniendo un estancamiento de la población y evitando una convergencia prematura.

La metodología propuesta constituye una modificación y adaptación de la idea propuesta por Pant et al. (2009). El algoritmo DE-PSO inicia con el usual algoritmo DE

hasta el punto donde el vector de ensayo es generado. Si éste vector es mejor que el correspondiente vector objetivo, entonces se incluye en la población, en otro caso, entra en el algoritmo la fase PSO y se genera un nuevo candidato solución usando las ecuaciones actualizadas de posición y velocidad de Cúmulo de Partículas, con la esperanza de encontrar una mejor solución. El método es repetido iterativamente hasta que se encuentre el valor óptimo (Thangaraj, R. et al.,2011). La inclusión de la fase PSO crea una perturbación en la población que ayuda a mantener diversidad de la población y a producir una solución óptima. El pseudocódigo del algoritmo híbrido DE-PSO se muestra la figura (3), allí w, c_1, c_2, r_1, r_2 son parámetros de control del cúmulo de partículas y P_i y P_g son las tasa de aprendizaje local y global, respectivamente. A su vez, F y C_r representan los parámetros de control factor de amplitud y la probabilidad de recombinación respectivamente para el algoritmo DE.

Con esta metodología se obtiene la estimación del vector de parámetros asociados a los efectos fijos y de los parámetros de la distribución asociado a los intersecciones aleatorias. En este trabajo sólo se considera la distribución log Gamma, sin embargo el algoritmo propuesto permite considerar otras distribuciones de probabilidad.

3.1. Algoritmo de estimación

El algoritmo de estimación implementado realiza un proceso iterativo combinando los métodos mencionados en la sección anterior. Éste parte del hecho de tener una base de datos constituida por medidas repetidas, con un vector de repuestas dicotómico y variables explicativas del mismo tipo. El criterio de convergencia empleado es el máximo número de iteraciones. Los pasos se describen a continuación:

1. Construir la función a minimizar (o función fitness) la cual debe tener como parámetros de entrada a los valores β_0, λ_0 y k_0 . En este caso la función corresponde a la suma de cuadrados del error:

```

Inicializar posición de partículas y vector velocidad
Do
For  $i=1$  to  $N$  (Tamaño de la Población) do
    Seleccionar  $r_1, r_2, r_3 \in N$  aleatoriamente
    //  $r_1, r_2, r_3$  son seleccionados tal que  $r_1 \neq r_2 \neq r_3$  //
    For  $j=1$  to  $D$  (Dimensión) do
        Seleccionar  $j_{rand} \in D$ 
        If ( $rand() < CR$  or  $j = j_{rand}$ )
            //  $rand()$  denota un número aleatorio distribuido
            // uniformemente entre 0 y 1 //
             $U_{ij,g+1} = x_{r_1,g} + F * (x_{r_2,g} - x_{r_3,g})$ 
        End if
    End for
    If ( $f(U_{i,g+1}) < f(X_{i,g})$ ) then
         $X_{i,g+1} = U_{i,g+1}$ 
    Else
        // Activa la fase PSO
        Encuentra una nueva partícula usando las ecuaciones actuali-
        zadas de posición y velocidad del Cúmulo de Partículas.
        Sea esta partícula  $TX = (tx_1, tx_2, \dots, tx_D)$  //
        For  $j = 1$  to  $D$  (dimensión) do
             $v_{ij,g+1} = w * v_{ij,g} + c_1 r_1 (P_{ij,g} - x_{ij,g}) + c_2 r_2 (P_{gj,g} - x_{ij,g})$ 
             $tx_{ij} = x_{ij,g} + v_{ij,g+1}$ 
        End for
        If ( $f(TX_i) < f(X_{i,g})$ ) then
             $X_{i,g+1} = TX_i$ 
        Else
             $X_{i,g+1} = X_{i,g}$ 
        End if
    End if
End for
Hasta alcanzar el criterio de parada

```

Figura 3.1: Pseudocódigo Algoritmo DE-PSO. Fuente: R. Thangaraj et al. (2011)

$$MSE = \sum_{i=1}^n (P(\mathbf{y}_i = 1) - \widehat{P}(\mathbf{y}_i = 1))^2$$

donde

$$P(\mathbf{y}_i = 1) = \prod_{j=1}^{J_i} \exp(-\exp(\alpha_i + X_{ij}\boldsymbol{\beta}))$$

2. Definir los valores de entrada para la parte de los algoritmos DE y PSO:

N: Número de partículas (individuos) en la población

p: número de parámetros a estimar

min.: Vector con los límites (bordes) inferiores de cada parámetro

max.: Vector con los límites (bordes) superiores de cada parámetro

Miter: Máximo número de iteraciones

f: Nombre de la función objetivo

CR: Probabilidad de recombinación

F.: Factor de mutación

φ_1 : Tasa de aprendizaje que controla el componente cognitivo

φ_2 : Tasa de aprendizaje que controla el componente social

w: factor inercia.

Hasta que el criterio de convergencia es alcanzado repita lo siguiente:

3. Genere uniformemente velocidades en $(-v_{max}, v_{max})$.
4. Use los límites superiores (\vec{b}_{up}) e inferiores (\vec{b}_{lo}) de cada parámetro para generar una población inicial aleatoria de estimaciones. Cada individuo de la población constituye un vector de la forma $\vec{x} = (\widehat{\beta}_0, \widehat{\beta}_1, \widehat{k}, \widehat{\lambda})$
5. Evalúe toda la población de estimaciones en la función fitness. Seleccione y guarde el mejor individuo (estimación de parámetros) en el vector x_{best} y su correspondiente evaluación en f_{best} .

6. Actualice el valor de cada individuo sumándole la velocidad hallada.
7. Inicie el algoritmo de evolución diferencial: Para cada individuo \vec{x} en la población
 - Seleccione aleatoriamente tres individuos \vec{a} , \vec{b} y \vec{c} de la población. Ellos deben ser distintos entre sí así como de \vec{x} .
 - Calcule el i -ésimo elemento del *potencial* nuevo individuo como sigue: $y_i = \begin{cases} a_i * F.(b_i - c_i), & \text{si } r_i < CR \\ x_i, & \text{e.o.c.} \end{cases}$ donde $r_i \sim U(0, 1)$.
 - Verifique las restricciones de borde: $y_i = \begin{cases} b_{lo_i}, & \text{si } y_i < b_{lo_i} \\ b_{up_i}, & \text{si } y_i > b_{up_i} \\ y_i, & \text{e.o.c.} \end{cases}$
 - Si $f(\vec{y}) < f(\vec{x})$ entonces actualice la posición $\vec{x} \leftarrow \vec{y}$.
8. En caso de que el nuevo individuo no produzca una evaluación del fitness mejor que \vec{x} aplique el algoritmo PSO:
 - Actualice las velocidades usando las tasas de aprendizaje y el factor de inercia.
 - Halle el nuevo individuo candidato, \vec{u} , sumándole la velocidad hallada.
 - Si $f(\vec{u}) < f(\vec{x})$ entonces actualice la posición $\vec{x} \leftarrow \vec{u}$. En caso contrario deje la posición original del individuo.

Este algoritmo fue implementado en el entorno de programación para análisis estadístico R y el código se puede ver en el apéndice B.

3.2. Prueba de convergencia del algoritmo

Los algoritmos PSO (global y local) y DE-PSO se probaron usando la función Schwefel 2.26 en el caso bidimensional. Se seleccionó esta función debido a su complejidad, pues es una función continua, no diferenciable, separable, escalable, multimodal

y simétrica. Esta función se define como

$$f(\mathbf{x}) = f[(x_1, \dots, x_d)] = 418,9829 d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|}) \quad -500 \leq x_i \leq 500 ; i = 1, \dots, d, \quad (3.2)$$

donde d es el número de variables.

En la siguiente tabla se resumen las características más representativas de la función (3.2).

Tabla 3.1: Características de la función Schwefel 2.26.

Tipo de problema	Minimización
Función fitness	$f(\mathbf{x}) = f[(x_1, \dots, x_n)] = 418,9829 d - \sum_{i=1}^d x_i \sin(\sqrt{ x_i })$
Dominio de búsqueda	$-500 \leq x_i \leq 500$
Mínimos locales	Varios mínimos locales
Mínimo global	$\mathbf{x}^* = (420,9687, \dots, 420,9687)$
Fitness óptimo	$f(\mathbf{x}^*) = 0$

Los valores de entrada que se utilizaron para el algoritmo de prueba fueron seleccionados de acuerdo a lo sugerido por Hvass(2010) para los algoritmos DE y PSO. En las tablas de resultados 3.2 y 3.3 se pueden ver los escenarios evaluados con su respectivo tamaño de población para un máximo de iteraciones $Miter = 100, 400, 4000$.

Para evaluar el desempeño del algoritmo PSO en sus dos versiones (global y local), se realizarón 50 corridas para cada número máximo de iteraciones. Los resultados se muestran a continuación en la tabla 3.2. En esta tabla, se muestran las estadísticas básicas para las evaluaciones de la función objetivo halladas en cada escenario. Las dos últimas columnas corresponde a los límites inferior (Li) y superior (Ls) de un intervalo al 95 % de confianza para el valor esperado (promedio) de la función fitness. En esta tabla la columna titulada *velocidad* hace referencia al método empleado para solucionar el problema del crecimiento de las velocidades: factor de inercia (w).

En la tabla 3.2 se observa que para la mayoría de los escenarios, el valor mínimo obtenido es cercano a 0 (cero), el valor mínimo global. Sin embargo los resultados del PSO Local, cuando $\varphi_1 = 2,8$ y $\varphi_2 = 1,3$ y valores de $N < 50$, son significativamente diferentes del valor mínimo global. Esto mismo ocurre para el PSO Global con $N = 10$ partículas.

Tabla 3.2: Resultados para PSO cuando se realizan 50 corridas para la función (3.2) bidimensional.

Tipo	φ_1	φ_2	Velocidad	N	Mínimo	Máximo	Media	Std.Dev(Media)	Li(95%)	Ls(95%)
Global	2.05	2.05	0,7298	10	$2,55E - 05$	776.4	297.5	0.8532	295.8	299.1
	2.05	2.05	0,7298	30	$2,55E - 05$	575.1	166.7	0.6478	165.4	168.0
	2.05	2.05	0,7298	50	$2,55E - 05$	438.3	104.0	0.51981	103.0	105.0
	2.8	1.3	0,7298	10	$5,22E + 01$	786.4	275.2	0.7339	273.7	276.6
	2.8	1.3	0,7298	30	$2,55E - 05$	463.6	143.6	0.6397	142.3	144.8
	2.8	1.3	0,7298	50	$2,55E - 05$	481.6	109.2	0.5389	108.2	110.3
	2.05	2.05	1,2	10	$3,84E - 05$	292.6	43.82	0.8844	42.1	45.6
	2.05	2.05	1,2	30	$2,76E - 05$	382.0	46.57	0.5705	45.4	47.7
	2.05	2.05	1,2	50	$2,57E - 05$	463.2	81.32	0.5351	80.3	82.4
	2.8	1.3	1,2	10	$3,34E - 05$	683.5	192.4	0.7613	190.9	193.9
	2.8	1.3	1,2	30	$2,60E - 05$	517.2	103.9	0.5748	102.8	105.0
2.8	1.3	1,2	50	$2,58E - 05$	472.3	71.26	0.4951	70.3	72.2	
Local	2.05	2.05	1,2	10	$2,98E - 05$	665.6	135.3	0.7127	133.9	136.7
	2.05	2.05	1,2	30	$2,61E - 05$	638.4	47.87	0.5262	46.8	48.9
	2.05	2.05	1,2	50	$2,56E - 05$	394.8	16.88	0.3279	16.2	17.5
	2.8	1.3	1,2	10	$5,03E - 05$	689.1	111.5	0.6962	110.1	112.8
	2.8	1.3	1,2	30	$2,99E - 05$	546.4	44.24	0.4869	43.3	45.2
	2.8	1.3	1,2	50	$2,95E - 05$	449.4	29.74	0.4404	28.9	30.6
	2.8	1.3	0,7298	10	$1,22E + 02$	710.5	384.9	0.8972	383.1	386.6
	2.8	1.3	0,7298	30	$1,45E + 01$	491.5	268.8	0.7631	271.0	274.0
	2.8	1.3	0,7298	50	$5,15E - 01$	504.2	220.5	0.6337	219.3	221.8

Efectivamente, se observa que el PSO global tiene una convergencia más rápida hacia el óptimo global. Sin embargo, cuando se utiliza *velocidad* 0,7298 y un número de partículas igual a 10, se puede estancar en un óptimo local (ver figura 3.2). Con respecto al algoritmo PSO local, se observa que la convergencia puede no ser tan rápida en algunos escenarios, sin embargo en general no se presenta el problema de los mínimos locales (ver figura 3.3).

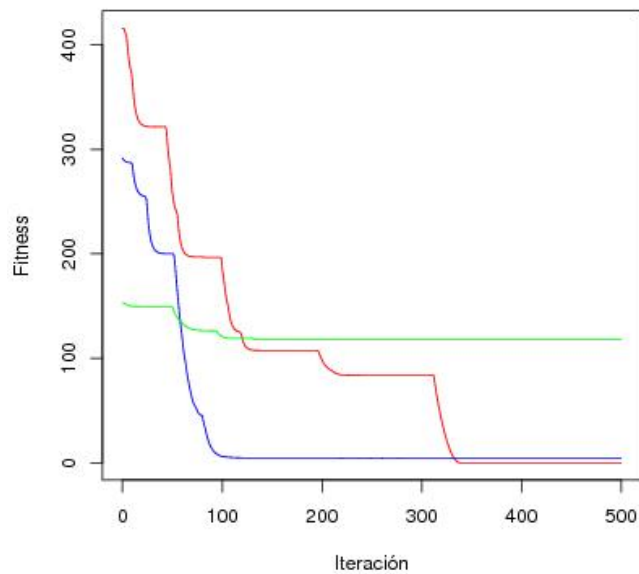


Figura 3.2: Desempeño del algoritmo PSO Global para optimizar la función Schwefel 2.26. Cada color corresponde al resultado de una corrida para máximo 500 iteraciones, $N = 50$ y $w = 0.7298$.

Para evaluar el desempeño del algoritmo DE, se realizaron 50 corridas para cada número máximo de iteraciones y tomando como parámetros los valores sugeridos por Hvass(2010). Los resultados se muestran a continuación en la tabla 3.3. En esta tabla, se muestran las estadísticas básicas para las evaluaciones de la función objetivo halladas en cada escenario. La penúltima columna muestra la desviación estándar de los valores mínimos de la función (3.2) y la última columna corresponde al intervalo al 95 % de confianza para el valor esperado (promedio) de la función fitness.

En la tabla 3.3 se observa que independientemente del escenario, el valor mínimo

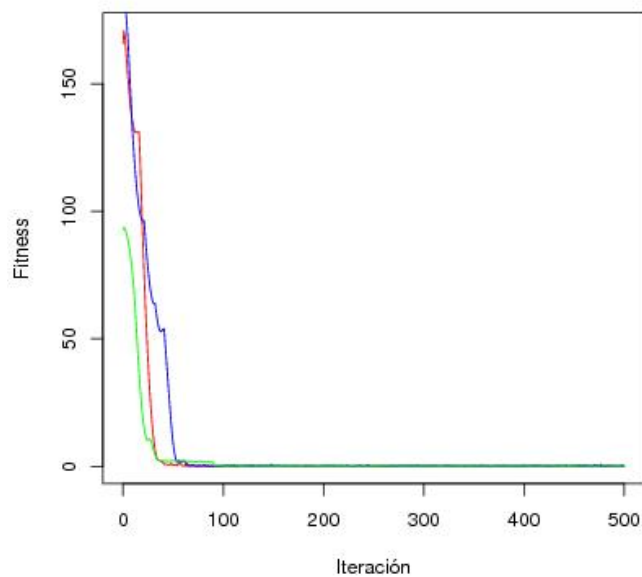


Figura 3.3: Desempeño del algoritmo PSO Local para optimizar la función Schwefel 2.26. Cada color corresponde al resultado de una corrida para máximo 500 iteraciones, $N = 50$ y $w = 0.7298$.

Tabla 3.3: Resultados para DE cuando se realizan 50 corridas para la función (3.2) bidimensional.

Iteraciones	N	CR	F	Mínimo	Máximo	Media	Std.Dev.	IC(95 %)
100	13	0.7450	0.9096	2.546e-05	6.084e+02	7.492e+01	9.494e+01	(72.2889, 77.5521)
400	13	0.7450	0.9096	2.546e-05	7.474e+02	4.606e+01	7.7069e+01	(44.9919, 47.1281)
400	10	0.4862	1.1922	2.546e-05	6.728e+02	8.912e+01	1.0427e+03	(87.6771, 90.5673)
4000	24	0.2515	0.8905	2.546e-05	5.711e+02	1.884e+01	5.5548e+01	(18.5966, 19.08345)
4000	20	0.7455	0.9362	2.546e-05	6.303e+02	1.187e+01	4.3191e+01	(11.6807, 12.05929)

corresponde a $2.546e-05$ que es aproximado a 0 (cero), el valor mínimo global. Así mismo, se ve que a medida que aumenta el número de iteraciones el valor medio y las desviaciones estándares de las evaluaciones de la función objetivo tienden a disminuir. Este comportamiento es más claro, en los intervalos de confianza cuyas amplitudes disminuyen a medida que aumenta el número de iteraciones.

Con respecto al algoritmo DE-PSO, la convergencia hacia el valor óptimo es muy rápida. Ésta ocurre aproximadamente en la iteración 30 indistintamente del escenario evaluado. En la figura (3.4) se ilustra este comportamiento en tres corridas distintas bajo el escenario 1, con $N = 13$ y $Miter = 400$.

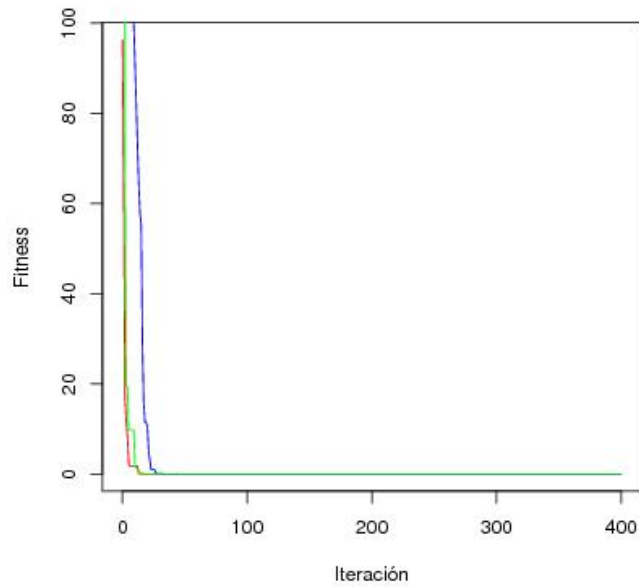


Figura 3.4: Desempeño del algoritmo DE-PSO para optimizar la función Schwefel 2.26. Cada color corresponde al resultado de una corrida para máximo 400 iteraciones.

Capítulo 4

Simulación

4.1. Modelo

Para el estudio se utilizó un modelo logístico mixto con intercepto aleatorio dado por:

$$P(y_{ij} = 1|\alpha_i) = \frac{\exp(\alpha_i - 4,8 + 2,3x_{ij} + e_{ij})}{1 + \exp(\alpha_i - 4,8 + 2,3x_{ij} + e_{ij})}; \quad i = 1, \dots, 50; \quad j = 1, \dots, 15. \quad (4.1)$$

donde x_{ij} una variable dicotómica fija, $\alpha_i \sim \text{logGamma}(50, 2)$ es el efecto aleatorio correspondiente a cada individuo. Éste valor es fijo para cada individuo pero varía entre individuos. La distribución seleccionada para los interceptos aleatorios fue la log Gamma con $k = 50$ y $\lambda = 2$. Por otra parte, e_{ij} es el término del error del i -ésimo individuo en la j -ésima medida repetida, y se asumió que su distribución es normal con media cero y desviación estándar 1.2.

4.2. Selección de parámetros

Para la implementación del algoritmo híbrido DE-PSO, es necesario definir los valores de los parámetros que determinan su comportamiento y eficacia en la optimización. En la literatura se pueden encontrar sugerencias para la elección de los valores de estos

parámetros en distintos escenarios tanto para el algoritmo PSO como DE; en esta propuesta se tomaron los valores sugeridos por Hvass, M. E. (2010) para ambos algoritmos. Los valores de los parámetros del algoritmo DE-PSO corresponden a una combinación de los valores sugeridos por el autor para el DE y PSO, teniendo en cuenta la dimensión del problema y del número de evaluaciones de la función objetivo.

El máximo número de iteraciones (*Miter*) y el tamaño de las poblaciones (*N*) se seleccionó de acuerdo a lo sugerido por el autor para el caso 4-dimensional: *Miter* =100,500,1000, y *N* =17, 47, 63.

Para el caso del algoritmo DE se evaluó un escenario, según los valores de la probabilidad de recombinación(*CR*) y el factor de mutación (*F*), a saber:

Escenario: *CR* =0.7122 y *F* =0.6301

Para el PSO, se fijó una velocidad máxima de $v_{max} = 1.2$ y se controlaron las tasas de aprendizaje cognitivo φ_1 y social φ_2 , y el factor de inercia ω , así:

Escenario 1: $\varphi_1 = -0.7238$, $\varphi_2 = 2.0289$ y $\omega = -0.3593$

Escenario 2: $\varphi_1 = -0.7238$, $\varphi_2 = 2.0289$ y $\omega = -0.1832$

Escenario 3: $\varphi_1 = -0.7238$, $\varphi_2 = 3.1913$ y $\omega = -0.3593$

Escenario 4: $\varphi_1 = -0.7238$, $\varphi_2 = 3.1913$ y $\omega = -0.1832$

Escenario 5: $\varphi_1 = 0.5287$, $\varphi_2 = 2.0289$ y $\omega = -0.3593$

Escenario 6: $\varphi_1 = 0.5287$, $\varphi_2 = 2.0289$ y $\omega = -0.1832$

Escenario 7: $\varphi_1 = 0.5287$, $\varphi_2 = 3.1913$ y $\omega = -0.3593$

Escenario 8: $\varphi_1 = 0.5287$, $\varphi_2 = 3.1913$ y $\omega = -0.1832$

4.3. Resultados

Es de resaltar que con el método propuesto en este trabajo, el interés radica en estimar el vector de parámetros asociados a los efectos fijos β y los parámetros de las

distribución que rige a los interceptos aleatorios: k y λ . En las tablas 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 y 4.9 se muestran los resultados de las simulaciones encontrados para cada combinación de escenarios DE, PSO, número de iteraciones y tamaños de las poblaciones. Cada celda contiene la siguiente información en el mismo orden:

- Valor estimado del parámetro correspondiente al menor valor fitness.
- Promedio de valores estimados (desviación estandar).
- Coeficiente de variación estimado (%).
- Intervalos formado por los valores mínimo y máximo hallados: (mínimo, máximo).

Las columnas 1-3, muestran los valores de los parámetros del PSO seleccionados. Las columnas 4 y 5 muestran los resultados de los parámetros del vector de efectos fijos β_0, β_1 ; las columnas 6 y 7 corresponden a los parámetros de la distribución Log-Gamma (K y L), y finalmente, la columna *Fitness* corresponde a los valores de función objetivo evaluada.

Tabla 4.1: Resultados numéricos con 1000 iteraciones para $N = 17$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4.8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	<i>Fitness</i>			
-0,7238	2,0289	-0,3593	-3.08303	1.23228	51.96249	5	2.50548e-7			
			-5.643(1.02)	3.424(0.90)	50.289(1.59)	2.897(1.52)	9.6e-6(2.31)			
			-18.12662	26.43328	3.16423	52.54955	24.06475			
			-6.8, -2.36	0.3, 4.3	48, 52	1, 5	2.5e-7, 1.74e-5			
	3,1913	-0,3593	-6.39174	4.08024	51.66131	2.41669	2.1949e-7			
			-5.674(0.99)	3.503(0.87)	50.032(1.65)	2.923(1.45)	9.88e-6(2.27e-6)			
0,5287	2,0289	-0,3593	-5.86247	4.3	49.01636	5	2.60959e-7			
			-5.189(0.88)	3.059(0.83)	50.167(1.27)	3.032(1.14)	9.34e-6(2.3e-6)			
			-17.00145	27.19536	2.52854	37.67953	24.84955			
			-6.8, -2.36	0.3, 4.3	48, 52	1, 5	2.6e-7, 1.7e-5			
	-0,1832	-5.57214	-4.983(0.83)	-16.66450	-6.8, 2.8	3.26935	48.02797	2.33067	1.75575e-7	
						2.918(0.79)	49.644(1.11)	3.217(1.12)	9.11e-6(2.41e-6)	
						27.15270	2.24493	34.95389	26.49203	
						0.3, 4.63	47.87, 52	1, 5	1.7e-7, 1.7e-5	
	3,1913	-0,3593	-5.98977	-5.569(0.95)	-17.02021	-6.8, -2.31	3.76145	50.11985	2.50438	5.4386e-7
							3.269(0.88)	50.106(1.42)	2.581(1.23)	9.8e-6(2.3e-6)
							26.82004	2.829705	47.60155	23.67601
							0.3, 4.3	48, 52	1, 5.72	5.4e-6, 1.7e-5
-0,1832	-6.8	-5.422(0.93)	-17.14478	-6.8, 0.93	4.3	48.88457	1.80735	1.41642e-7		
					3.205(0.86)	50.060(1.37)	2.773(1.19)	9.60e-6(2.4e-6)		
					26.89259	2.74431	42.86317	24.57197		
					0.3, 4.3	47.8, 52.7	1, 5.8	1.4e-7, 1.7e-5		

Tabla 4.2: Resultados numéricos con 500 iteraciones para $N = 17$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4,8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	<i>Fitness</i>
-0,7238	2,0289	-0,3593	-5.50837	4.3	48.08396	5	1.75678e-6
			-5.771(1.0)	3.744(0.70)	50.015(1.65)	2.592(1.57)	1.12e-4(7.14e-5)
			-17.36748	18.71086	3.29136	60.76033	63.48683
			-6.8, 2.44	0.49, 4.3	48, 52	1, 5.23	1.7e-6, 4.8e-4
	-0,1832	-0,1832	-5.52298	4.3	51.96459	5	4.877e-6
			-5.627(1.07)	3.6275(0.78)	50.095(1.71)	2.785(1.72)	1.099e-4(7.0e-5)
			-18.95659	21.45519	3.41492	61.74947	63.79681
			-6.8, 2.36	0.66, 4.3	48, 52	1, 5	4.88e-6, 4.8e-4
	3,1913	-0,1832	-5.52298	4.3	51.96459	5	4.87743
			-5.627(1.07)	3.627(0.78)	50.095(1.71)	2.785(1.72)	1.09e-4(7.01e-5)
			-18.95659	21.45519	3.41492	61.74947	63.79681
			-6.8, 2.36	0.66, 4.3	48, 52	1, 5	4.9e-6, 4.8e-4
0,5287	2,0289	-0,3593	-5.95455	4.3	48.93299	3.45047	1.7574e-6
			-5.483(0.83)	3.529(0.62)	50.059(1.21)	2.642(1.22)	1.089e-4(6.9e-5)
			-15.09184	17.68488	2.42308	46.14044	63.83410
			-6.8, -1.62	0.31, 4.3	47.9, 52	0.51, 5	1.7e-6, 4.8e-4
	-0,1832	-0,1832	-5.61804	4.27690	51.46249	4.05448	3.3661e-6
			-5.263(0.78)	3.476(0.67)	49.933(1.21)	3.120(1.26)	1.04e-4(6.58e-5)
			-14.88004	19.30082	2.43011	40.43573	63.31749
			-6.8, -2.73	0.5, 4.57	48, 52	1, 5	3.4e-6, 4.8e-4
	3,1913	-0,1832	-6.46183	3.74870	51.40388	1.16000	2.66047e-6
			-5.900(0.77)	3.701(0.60)	49.779(1.42)	1.998(0.10)	1.12e-4(7.04e-5)
			-12.98275	15.92395	2.84856	49.96639	63.09041
			-6.8, -2.96	0.94, 4.3	48, 52	1, 5.65	2.7e-6, 4.8e-4

Tabla 4.3: Resultados numéricos con 100 iteraciones para $N = 17$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4,8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	$Fitness$	
-0,7238	2,0289	-0,3593	-4.92103	3.02744	49.49098	3.26234	3.3255e-5	
			-5.403(0.98)	3.343(0.93)	50.051(1.52)	2.939(1.3)	6.34e-4(3.6e-4)	
			-18.23898	27.75499	3.048664	44.63463	56.72706	
			-6.8, -0.54	0.3, 4.3	48, 52	1, 5	3.3e-5, 0.0013	
	-0,1832	-0,1832	-0,1832	-4.46559	2.48258	50.34364	3.40895	2.2939e-5
				-5.685(0.85)	3.536(0.81)	50.405(1.58)	2.776(1.45)	6.08e-4(3.6e-4)
				-14.95621	23.03794	3.13785	52.26386	59.50667
				-6.8, -2.69	0.3, 4.3	48, 52	1, 5	2.3e-5, 0.0013
	3,1913	-0,1832	-0,1832	-4.46559	2.48258	50.34364	3.40895	2.2939e-5
				-5.685(0.85)	3.536(0.81)	50.405(1.58)	2.776(1.45)	6.08e-4(3.6e-4)
				-14.95621	23.03794	3.13785	52.26386	59.50667
				-6.8, -2.69	0.3, 4.3	48, 52	1, 5	2.3e-5, 0.0013
0,5287	2,0289	-0,3593	-4.82344	3.05016	49.20963	3.89519	2.796e-5	
			-5.297(0.95)	3.207(0.84)	49.766(1.29)	2.792(1.08)	6.03e-4(3.6e-4)	
			-17.92786	26.16217	2.59763	38.83097	60.53111	
			-6.8, -1.73	0.3, 4.3	48, 52	0.81, 5	2.8e-5, 0.0013	
	-0,1832	-0,1832	-0,1832	-3.96560	2.39799	49.67213	4.80089	2.777e-5
				-4.895(0.87)	2.968(0.90)	50.470(1.15)	3.370(1.07)	6.03e-4(3.6e-4)
				-17.69545	30.37008	2.28842	31.81893	60.88901
				-6.8, 0.73	0.3, 4.32	48, 52	1, 5	2.8e-5, 0.0013
	3,1913	-0,1832	-0,1832	-5.68635	4.3	49.46450	5	5.0086e-5
				-5.280(1.1)	3.197(0.10)	49.720(1.29)	2.962(1.36)	6.86e-4(3.7e-4)
				-20.78975	31.23893	2.59917	45.89008	54.19305
				-6.8, -2.64	0.35, 4.3	48, 52.35	1, 5	5.0e-5, 0.0013

Tabla 4.4: Resultados numéricos con 1000 iteraciones para $N = 47$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4,8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	$Fitness$
-0.7238	2,0289	-0.1832	-6.72284	4.08564	48.05506	1.20024	1.48399e-6
			-5.445(1.26)	3.224(1.06)	49.927(1.64)	2.355(1.51)	7.65e-6(1.5e-6)
			-23.22561	33.03618	3.29370	64.24001	20.02242
	3,1913	-0,3593	-6.8, -2.31	0.3, 4.3	48, 52	0.95, 5.36	1.5e-6, 1.3e-5
			-6.10219	4.3	48.79728	2.69222	1.62945e-6
			-5.764(1.02)	3.443(0.93)	49.904(1.55)	1.921(1.09)	7.83e-6(1.5e-6)
3,1913	-0,1832	-17.6839	27.0247	3.1102	56.6395	19.6823	
		-6.8, -2.4	0.3, 4.3	48, 52	0.9, 5	1.6e-6, 1.3e-5	
		-6.72285	4.08565	48.05506	1.20024	1.48399e-6	
0.5287	2,0289	-0,3593	-5.445(1.26)	3.224(1.06)	49.927(1.64)	2.355(1.51)	7.65e-6(1.5e-6)
			-23.22561	33.03618	3.29370	64.24001	20.02242
			-6.8, -2.31	0.3, 4.3	48, 52	0.95, 5.36	1.5e-6, 1.3e-5
	3,1913	-0,3593	-5.18673	2.95509	48.36015	1.99219	1.45987e-6
			-4.688(0.9)	2.712(0.85)	50.236(1.21)	2.904(1.13)	7.39e-6(1.6e-6)
			-19.23151	31.31224	2.41848	39.06062	21.13738
3,1913	-0,1832	-6.8, -2.35	0.3, 4.3	48, 52	1, 5	1.4e-6, 1.3e-5	
		-6.20673	3.88133	49.61495	1.68111	1.46606e-6	
		-5.452(1.02)	3.21(0.93)	49.896(1.45)	2.115(1.07)	7.79e-6(1.6e-6)	
3,1913	-0,3593	-18.68930	29.14509	2.90194	50.80916	20.27198	
		-6.8, -2.33	0.3, 4.3	48, 52	1, 5	1.5e-6, 1.3e-5	
		-6.8	4.3	49.97714	1.50001	1.74061e-6	
3,1913	-0,1832	-5.33(1.01)	3.086(0.92)	49.925(1.35)	2.137(1.03)	7.6e-6(1.56e-6)	
		-18.99885	29.98787	2.71111	48.23675	20.59556	
		-6.8, -2.31	-0.12, 4.3	48, 52.29	1, 5	1.74e-6, 1.33e-5	

Tabla 4.5: Resultados numéricos con 500 iteraciones para $N = 47$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4,8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	$Fitness$	
-0,7238	2,0289	-0,3593	-6.71804	4.3	52	1.55741	6.65418e-6	
			-5.907(0.69)	3,859(0.46)	50.126(1.52)	2.716(1.40)	7.737e-4(0.003)	
			-11.71846	12.06190	3.03839	51.58616	360.4919	
			-6.8, -3.44	0.89, 4.3	48, 52	1, 5	6.65e-6, 0.0698	
	-0,1832	-0,1832	-0,1832	-5.17059	3.26561	51.93671	3.11461	1.06367e-5
				-5.839(0.75)	3.848(0.48)	50.123(1.68)	2.937(1.54)	6.39e-4(0.0018)
				-12.90885	12.49479	3.34672	52.40549	289.369
				-6.8, -3.5	1.47, 4.3	48, 52	1, 5	1.06e-6, 0.070
	3,1913	-0,3593	-0,3593	-6.08814	3.54592	49.29707	1.68975	2.07543e-5
				-5.982(0.72)	3.878(0.50)	50.289(1.52)	2.572(1.37)	1.16e-3(0.004)
				-12.10917	12.96325	3.03247	53.16825	329.382
				-6.8, -3.45	1.16, 4.3	48, 52	1, 5	2.07e-5, 0.068
-0,1832	-0,1832	-0,1832	-5.17059	3.26562	51.93671	3.11461	1.064e-5	
			-5.839(0.75)	3.848(0.48)	50.123(1.68)	2.937(1.54)	6.39e-4(0.002)	
			-12.90885	12.49479	3.34672	52.40549	289.369	
			-6.8, -3.5	1.47, 4.3	48, 52	1, 5	1.06e-5, 0.07	
0,5287	2,0289	-0,1832	-5.24164	3.87655	51.41547	4.88234	5.29122e-6	
			-5.821(0.58)	3.772(0.39)	50.104(1.24)	2.619(1.11)	6.26e-4(0.002)	
			-10.00145	10.27619	2.48327	42.55240	348.424	
			-6.8, -3.72	1.67, 4.3	48, 52	1, 5	5.29e-6, 0,07	
	3,1913	-0,3593	-0,3593	-6.36047	4.09182	50.5204	2.03988	5.88169e-6
				-6.031(0.67)	3.847(0.45)	50.124(1.46)	2.342(1.23)	7.94e-4(0.002)
				-11.08547	11.84832	2.9284	52.35749	304.824
				-6.8, -3,08	1.036, 4.3	48, 52.08	0.88, 5	5.9e-6, 0.067
	-0,1832	-0,1832	-0,1832	-5.65546	4.29986	49.09785	4.82151	1.26656e-5
				-5.986(0.64)	3.850(0.42)	50.121(1.4)	2.433(1.21)	7.03e-4(0.002)
				-10.69410	10.87102	2.79121	49.63609	341.7877
				-6.8, -2.51	0.51, 4.58	48, 52	1, 5	1.3e-5, 0.071

Tabla 4.6: Resultados numéricos con 100 iteraciones para $N = 47$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4,8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	<i>Fitness</i>		
-0,7238	2,0289	-0,1832	-5.56659	4.02763	50.35531	4.10415	1.07333e-5		
			-5.713(0.70)	3.770(0.53)	50.130(1.55)	3.023(1.44)	0.001(0.004)		
			-12.31644	14.01873	3.08410	47.53493	420.184		
			-6.8, -3.78	1.59, 4.3	48, 52	1, 5	1.07e-5, 0.07		
	3,1913	-0,3593	-6.08814	3.5459	49.29707	1.68975	2.0754e-5		
			-5.982(0.72)	3.878(0.50)	50.289(1.52)	2.572(1.37)	1.16e-3(0.004)		
-0,1832	-0,1832	-0,1832	-12.10917	12.96325	3.03247	53.16825	329.382		
			-6.8, -3.45	1.16, 4.3	48, 52	1, 5	2.1e-5, 0.07		
			-5.56659	4.02762	50.35531	4.10415	1.07333e-5		
			-5.713(0.70)	3.770(0.53)	50.130(1.55)	3.023(1.44)	0.001(0.004)		
0,5287	2,0289	-0,1832	-12.31644	14.018731	3.084097	47.53493	420.184		
			-6.8, -3.78	1.59, 4.3	48, 52	1, 5	1.07e-5, 0.07		
			-4.95181	3.08278	50.45836	2.89671	1.57268e-5		
			-5.548(0.60)	3.710(0.50)	50.368(1.18)	3.214(1.16)	0.001(0.005)		
-10.86551	13.57427	2.34124	36.15012	462.687	1.57e-5, 0.07				
						-6.8, -3.50	0.77, 4.3	48, 52	1, 5

Tabla 4.7: Resultados numéricos con 1000 iteraciones para $N = 63$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4,8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	<i>Fitness</i>
-0,7238	2,0289	-0,3593	-6.16014	3.79284	49.37863	2.16991	1.7454e-7
			-5.313(1.03)	3.064(0.99)	50.012(1.57)	2.878(1.35))	7.505e-6(2.3e-6)
			-19.39813	32.23621	3.14912	46.85565	31.41800
			-6.8,-2.26	0.3, 4.3	48, 52	1, 5	1.7e-7, 1.7e-5
	3,1913	-0,3593	-6.39174	4.08024	51.66131	2.41669	2.1949e-7
			-5.674(0.10)	3.503(0.87)	50.032(1.65)	2.923(1.45)	9.88e-6(2.3e-6)
0,5287	2,0289	-0,3593	-5.86247	4.3	49.01637	5	2.6096e-7
			-5.189(0.88)	3.059(0.83)	50.167(1.27)	3.032(1.14)	9.34e-6(2.3e-6)
			-17.00145	27.19536	2.52854	37.67953	24.84955
			-6.8, -2.36	0.3, 4.3	48, 52	1, 5	2.6e-7, 1.7e-5
		-0,1832	-5.57214	3.26935	48.02797	2.33067	1.7557e-7
			-4.983(0.83)	2.918(0.79)	49.644(1.11)	3.217(1.12)	9.11e-6(2.4e-6)
	3,1913	-0,3593	-16.66450	27.15270	2.24493	34.95389	26.49203
			-6.8, 2.8	0.3, 4.63	47.87, 52	1, 5	1.7e-7, 1.7e-5
			-5.98977	3.76145	50.11985	2.50438	5.4386e-7
			-5.569(0.95)	3.269(0.88)	50.106(1.42)	2.581(1.23)	9.802e-6(2.3e-6)
		-0,1832	-17.02021	26.82004	2.82970	47.60155	23.67601
			-6.8, -2.31	0.3, 4.3	48, 52	1, 5.72	5.4e-7, 1.7e-5
-0,1832	-6.8	4.3	48.88457	1.80735	1.4164e-7		
	-5.422(0.93)	3.205(0.86)	50.060(1.37)	2.773(1.19)	9.6e-6(2.4e-6)		
	-17.14478	26.89259	2.74431	42.86317	24.57197		
	-6.8, 0.93	0.3, 4.3	47.8, 52.7	1, 5.8	1.4e-7, 1.7e-5		

Tabla 4.8: Resultados numéricos con 500 iteraciones para $N = 63$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4,8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	<i>Fitness</i>
-0,7238	2,0289	-0,1832	-6.8	4.09373	48.03991	1	2.825e-6
			-4.696(1.17)	2.760(1.03)	49.856(1.49)	3.120(1.37)	7.33e-6(1.5e-6)
			-24.85022	37.36788	2.98979	43.89558	20.91351
	3,1913	-0,3593	-6.8, -2.35	0.3, 4.3	48, 52.7	1, 5	2.8e-6, 1.3e-5
			-6.8	3.99530	50.24586	1	1.577e-6
			-5.808(0.99)	3.469(0.91)	49.829(1.53)	1.872(1.03)	7.47e-6(1.5e-6)
3,1913	-0,1832	-16.87151	26.13864	3.07486	55.10451	20.14689	
		-6.8, -2.53	0.3,4.3	48, 52	1, 5	1.6e-6, 1.3e-5	
		-6.8	4.09372	48.03991	1	2.825e-6	
0,5287	2,0289	-0,3593	-4.696(1.17))	2.760(1.03)	49.856(1.49)	3.120(1.37)	7.329e-6(1.5e-6)
			-24.85022	37.36788	2.98979	43.89558	20.91351
			-6.8, -2.35	0.3, 4.3	48, 52.7	1, 5	2.8e-6, 1.3e-5
	3,1913	-0,1832	-4.77904	2.95574	50.22418	2.86718	2.306e-6
			-4.607(0.96)	2.653(0.90)	50.026(1.22)	3.019(1.21)	7.203e-6(1.5e-6)
			-20.82211	34.11721	2.44107	40.18035	21.43601
3,1913	-0,1832	-6.98, -2.57	0.3, 4.7	48, 52	0.54, 5	2.3e-6, 1.3e-5	
		-6.21635	4.04612	50.06615	1.90913	1.640e-6	
		-5.218(0.1)	3.062(0.91)	49.943(1.34)	2.357(1.11)	7.272e-7(1.5e-6)	
3,1913	-0,1832	-19.09789	29.59190	2.68517	47.240421	20.99144	
		-6.8, -2.53	0.3, 4.3	48, 52	1, 5	1.6e-6, 1.3e-5	

Tabla 4.9: Resultados numéricos con 100 iteraciones para $N = 63$ individuos, $CR = 0,7122$ y $F = 0,6301$

φ_1	φ_2	ω	$\beta_0(-4,8)$	$\beta_1(2,3)$	$K(50)$	$L(2)$	<i>Fitness</i>
-0,7238	2,0289	-0,1832	-6.8	4.09373	48.03991	1	2.825e-6
			-4.696(1.17)	2.760(1.03)	49.856(1.49)	3.120(1.37)	7.329e-6(1.5e-6)
			-24.85022	37.36788	2.98979	43.89558	20.91351
			-6.8, -2.35	0.3, 4.3	48, 52.7	1, 5	2.8e-6, 1.3e-5
	3,1913	-0,3593	-5.65863	3.11321	52	1.52074	2.0729e-6
			-5.394(1.13)	3.219(1)	49.886(1.57)	2.344(1.31)	7.651e-6(1.6e-6))
			-21.02811	31.24628	3.14243	55.83467	21.48986
			-6.8, -2.53	0.35, 4.3	47.6, 52	1, 5.31	2.1e-6, 1.3e-5
		-0,1832	-6.8	4.09373	48.03991	1	2.82522
			-4.696(1.17)	2.760(1.03)	49.856(1.49)	3.120(1.37)	7.329e-6(1.5e-6)
			-24.85022	37.36788	2.98979	43.89558	20.91351
			-6.8, 2.35	0.3, 4.3	48, 52.7	1, 5	2.8e-6, 1.3e-5
0,5287	2,0289	-0,3593	-4.7790	2.95574	50.22418	2.86718	2.3061e-6
			-4.607(0.96)	2.653(0.90)	50.026(1.22)	3.019(1.21)	7.203e-6(1.5e-6)
			-20.82211	34.11721	2.44107	40.18035	21.43601
			-6.98, -2.57	0.3, 4.7	48, 52	0.54, 5	2.3e-6, 1.3e-5

En general, se observa que en el algoritmo la elección de los parámetros asociados a los algoritmos PSO y DE son fundamentales. Así como la elección del número máximo de iteraciones e individuos. Es por ello, que en este tipo de técnicas metaheurísticas y en particular para la estimación de éste modelo, no se cumple literalmente el hecho de que a medida que aumentan el número de iteraciones mejoran las estimaciones; en los resultados, se nota que hay escenarios con pocas iteraciones que muestran un mejor comportamiento. Además, con los resultados obtenidos se resalta la importancia de realizar distintas corridas en cada escenario a fin de examinar la homogeneidad de las estimaciones obtenidas.

Nótese que en todos los escenarios evaluados y parámetros estimados, las estimaciones halladas se caracterizan porque el intervalo *estimacion promedio*, \pm *desviacion* siempre contiene el valor real. En particular para cada parámetro estimado se obtuvo lo siguiente:

- La estimación de los parámetros de efectos fijos β_0 y β_1 siempre se dieron con el signo indicado (negativo para el primero y positivo para el segundo). Esto indica que con este mecanismo de estimación se puede detectar el tipo de asociación entre las variables explicativas y la de respuesta. Adicionalmente, se observa que las estimaciones obtenidas en cada escenario evaluado, fueron homogéneas.
- Independientemente del escenario, el parámetro asociado a la distribución del intercepto aleatorio K , fue el *mejor estimado*, con un coeficiente de variación inferior al 4% en todos los casos. Por otra parte, si bien el parámetro λ presentó unas estimaciones heterogéneas, en cada escenario considerado, se observa que las estimaciones mejoran al aumentar el número de iteraciones e individuos.

Los mejores resultados se dieron en las simulaciones de $N = 17$ individuos con 100 iteraciones, específicamente los escenarios 2 ($\varphi_1 = -0.7238$, $\varphi_2 = 2.0289$, $\omega = -0.1832$) y 4 ($\varphi_1 = -0.7238$, $\varphi_2 = 3.1913$, $\omega = -0.1832$), que muestran resultados similares, son las estimaciones más cercanas al valor real, teniendo en cuenta los cuatro parámetros en conjunto.

Por otra parte, es de resaltar la convergencia que presentó el algoritmo híbrido (ver figuras 4.3 y 4.3), pues se ve que en pocas iteraciones los valores comienzan a estabilizarse, lo que una vez más comprueba que un número pequeño de iteraciones da confiabilidad en las estimaciones.

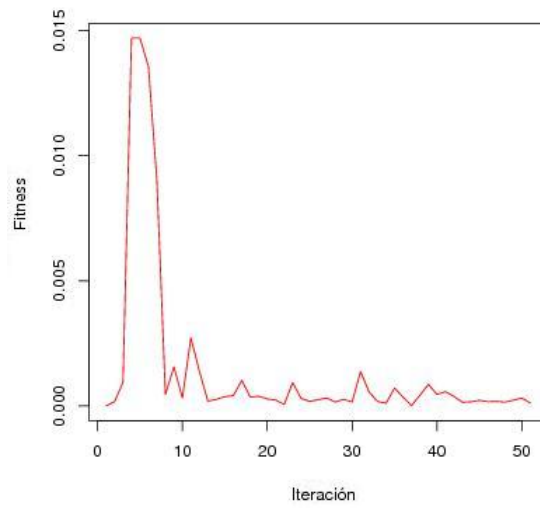


Figura 4.1: Convergencia de una sola corrida

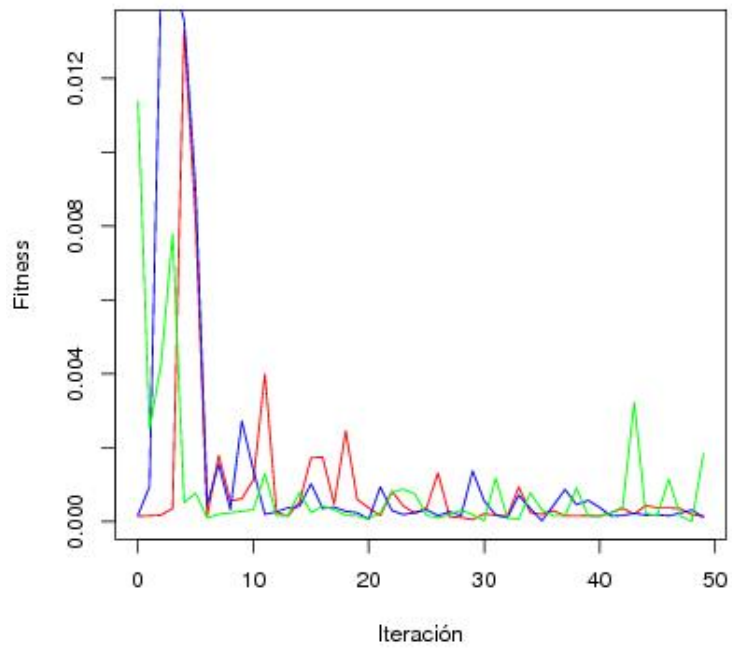


Figura 4.2: Convergencia de tres corridas. Cada color representa una corrida.

4.4. Comparación con un método convencional

Algunas de las metodologías tradicionales de estimación han sido implementadas en entornos de programación estadísticos. Para el caso del Modelo Logístico Mixto, el lenguaje de programación R ofrece un paquete que puede utilizarse para ajustar el modelo y obtener la estimación de sus parámetros, utilizando un método de estimación convencional. El paquete se denomina *glmmPQL* y esta disponible en la *librería (MASS)*, que permite ajustar un Modelo Lineal Mixto Generalizado usando Quasi-Verosimilitud Penalizada.

Con el objeto de hacer una comparación de las estimaciones obtenidas a través del algoritmo implementado en este trabajo, se tomó el modelo (4.1) que fue ajustado para la simulación en la primera sección de este capítulo, conservando el tamaño de la población y los valores originales de los parámetros y se utilizó el paquete *glmmPQL* para obtener la estimación de los parámetros. Se implementó un algoritmo con el que inicialmente se genera una base de datos usando los parámetros definidos en la sección 4.1 y luego se invoca la instrucción de ajuste del modelo para obtener la estimación de los parámetros β_0 y β_1 con sus respectivas errores estándar. Es importante aclarar que cada vez que se corre la instrucción de ajuste para la misma base de datos, las estimaciones obtenidas no varían, aún cambiando la semilla de generación de los números pseudoaleatorios. Es por ello, que para examinar el comportamiento del método se generaron 50 bases de datos con las características mencionadas anteriormente; los resultados se muestran en la siguiente tabla.

Con los resultados mostrados en la tabla 4.10, se observa lo siguiente:

- Todas las corridas muestran la misma tendencia en los resultados.
- Si bien el método convencional produce un resultado en menor tiempo, éste no es capaz de estimar correctamente los parámetros del modelo. Los intervalos de confianza que se pueden formar a partir de las estimaciones y sus respectivas desviaciones estándar, nunca contendrán el verdadero valor del parámetro.
- En cambio el método propuesto en este trabajo, basado en técnicas metaheurísti-

Tabla 4.10: Resultados obtenidos con el paquete glmmPQL de R para $N = 50$ individuos, $\beta_0 = -4,8$ y $\beta_1 = 2,3$

β_0		β_1		β_0		β_1	
Estimación	Error Std.	Estimación	Error Std.	Estimación	Error Std.	Estimación	Error Std.
1.534849	0.1379966	-3.568828	0.2103310	1.373222	0.1277500	-3.322233	0.2030307
1.528516	0.1418262	-3.208694	0.1913074	1.418377	0.1353613	-3.306283	0.1977205
1.575988	0.1473243	-3.413573	0.1977965	1.370742	0.1245868	-3.234327	0.2004108
1.429276	0.1295964	-3.558728	0.2134491	1.288318	0.1292182	-3.045346	0.1895192
1.535543	0.1398770	-3.411903	0.1990947	1.453096	0.1367532	-3.446148	0.2024804
1.572960	0.1344236	-3.285938	0.1989658	1.542455	0.1309285	-3.190427	0.1961958
1.358962	0.1364060	-3.044901	0.1884519	1.520985	0.1373715	-3.349271	0.1991082
1.628955	0.1407589	-3.594556	0.2075323	1.581945	0.1431374	-3.585674	0.2102228
1.372501	0.1312127	-3.197648	0.1969209	1.466337	0.1309188	-3.220356	0.1972852
1.474474	0.1409300	-3.299636	0.1944612	1.342654	0.1349751	-3.058255	0.1876573
1.513135	0.1580153	-3.745858	0.2074802	1.500621	0.1371887	-3.469355	0.2041961
1.249290	0.1254713	-3.053337	0.1932839	1.342340	0.1344546	-3.268641	0.1963280
1.370742	0.1245868	-3.234327	0.2004108	1.288318	0.1292182	-3.045346	0.1895192
1.535543	0.1398770	-3.411903	0.1990947	1.453096	0.1367532	-3.446148	0.2024804
1.572960	0.1344236	-3.285938	0.1989658	1.542455	0.1309285	-3.190427	0.1961958
1.358962	0.1364060	-3.044901	0.1884519	1.520985	0.1373715	-3.349271	0.1991082
1.628955	0.1407589	-3.594556	0.2075323	1.581945	0.1431374	-3.585674	0.2102228
1.348172	0.1268803	-3.536991	0.2142690	1.342340	0.1344546	-3.268641	0.1963280
1.249290	0.1254713	-3.053337	0.1932839	1.500621	0.1371887	-3.469355	0.2041961
1.513135	0.1580153	-3.745858	0.2074802	1.342654	0.1349751	-3.058255	0.1876573
1.276631	0.1273638	-3.178880	0.1970489	1.534112	0.1481467	-3.451597	0.1970414
1.353718	0.1271806	-2.995946	0.1900541	1.446066	0.1339911	-3.311934	0.2005491
1.433761	0.1313245	-3.115125	0.1934001	1.399858	0.1309336	-3.179381	0.1960348
1.449333	0.1302283	-3.439695	0.2070293	1.319805	0.1383462	-3.325295	0.1970461
1.523175	0.1479391	-3.222124	0.1907176	1.544787	0.1364381	-3.732899	0.2159008

cas, como se mostró en la sección anterior, si permite obtener una mejor estimación de ellos; a pesar de que requiere el uso de la exploración de varios escenarios.

El algoritmo utilizado para obtener los resultados de esta sección se puede ver en apéndice B.

Capítulo 5

Conclusiones

Los modelos lineales mixtos generalizados proveen una gran flexibilidad para tratar modelos lineales clásicos con niveles múltiples de efectos aleatorios incorporados en varias categorías. El análisis de datos continuos siempre ha estado asociado a los modelos lineales mixtos y con la extensión de estos modelos a su forma generalizada, este análisis logró alcanzar recientemente un desarrollo teórico unificado y ha abierto una gran ventana a la investigación. El modelo Logístico Mixto se convirtió en uno de los desarrollos más importante dentro de los modelos GLMM, parece ser la herramienta con que se abordará en el futuro la modelación de las situaciones de elección, gracias a su potencia y flexibilidad a la hora abordar tanto las posibles heteroscedasticidades como las correlaciones entre alternativas.

A pesar de la potencia que ha tomado el modelo Logístico Mixto en la investigación y en la práctica, la variada literatura que lo respalda deja ver una particularidad en el desarrollo del modelo que podríamos tomar como una debilidad en la utilización del mismo, la intervención de métodos numéricos que agregan una alta carga computacional en la estimación de los parámetros del modelo representa la mayor dificultad en la óptima aplicación del modelo Logístico Mixto. La especificación del modelo a través de la función de enlace *Logit*, lleva el desarrollo de la estimación a problemas numéricos en la solución analítica de casos de integrabilidad.

La estimación de un Modelo Logístico Mixto se convierte entonces en un problema donde la dificultad se presenta principalmente por la alta complejidad de los cálculos y la duración de éstos. La diversidad de este tipo de problemas en todos los campos, ha provocado el desarrollo de técnicas metaheurísticas, que son algoritmos aproximados de optimización y búsqueda de propósito general, los cuales se pueden aplicar a diferentes problemas de optimización con mínimos cambios para ser adaptado a un problema específico.

Gracias a que las técnicas metaheurísticas son capaces de proporcionar muy buenas soluciones (no necesariamente la óptima pero si aproximada) en tiempo y con recursos razonables, se convierten en un herramienta útil para dar un enfoque mejorado a la estimación del modelo Logístico Mixto. De la combinación del uso de la función de linkeo *Log-Log* y la metaheurística del algoritmo híbrido DE-PSO, resulta una óptima solución para la estimación de los parámetros del modelo Logístico Mixto con intercepto aleatorio, como lo pudimos ver en los resultados arrojados por las simulaciones, en el capítulo 4. Con esta propuesta de estimación se logra encontrar la solución analítica de los problemas de integrabilidad que plantea el desarrollo habitual de estimación usando la función de linkeo *logit*, y se logran estimaciones más cercanas a los parámetros que las obtenidas con los métodos convencionales.

Bibliografía

- [1] Agresti, A., Booth, J., Hobert, J.P. and Caffo, B.(2000). *Random effects modeling of categorical response data*. Sociological Methodology, 30, 27-80.
- [2] Agresti, A.(2002). *Categorical Data Analysis*. New York: Wiley, 2nd edition.
- [3] Bock, R.D. and Aitkin, M.(1981). *Marginal Maximum Likelihood Estimation of Item Parameters: Application Of An EM Algorithm*. Psychometrika 46, 443-459.
- [4] Breslow, N.E. (1989). *Score tests in overdispersed GLMs*. Springer-Verlag, New York, pages 64-74.
- [5] Breslow, N.E. and Clayton, D.G.(1993). *Approximate inference in generalized linear mixed models*. Journal American Statistical Association, 88, 9-25.
- [6] Brown, H. and Prescott, R. (2001). *Applied Mixed Models in Medicine*. New York, John Wiley & Sons, Ltd.
- [7] Casella, G. and Berger, R.L. (2002). *Statistical Inference. 2nd Edition*. Pacific Grove, CA. Duxbury.
- [8] Chen, Z. and Kuo, L. (2001). *A Note on the Estimation of the Multinomial Logit Model with Random Effects*. The American Statistician 55, 89-95.
- [9] Chintagunta, P., Jain, D. and Vilcassim, N. (1994). *A Random-Coefficients Logit Brand Choice Model Applied to Panel Data*. Journal of Business and Economic Statistics, 12 (3), pp. 317-328.

- [10] Chintagunta, P., Jain, D. and Vilcassim, N. (1994). *Heterogeneity in Brand Preferences in Logit Models for Panel Data*. Journal of Marketing Research, pp. 417-28.
- [11] Conaway, M.R. (1990). *A Random Effects Model for Binary Data*. Biometrics 46, 317-328.
- [12] Cook, R.D. and Weisberg, S. (1989). *Residual and influence in regression*. Ed. Chapman and Hall.
- [13] Davidian, M. and Giltinan, D. (1998). *Nonlinear Models for Repeated Measurement Data*. New York, Chapman & Hall/CRC.
- [14] Dean, C.B. and Nielsen, J.D. (2007). *Generalized linear mixed model: a review and some extensions*. Springer Sciences+Business Media 13, 497-512.
- [15] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society Series B, 39(1), 1-38.
- [16] Fress, E.W.(2004). *Longitudinal and Panel Data: analysis and applications in the social sciences*. Cambridge University Press.
- [17] García, J.M.(2006). *Algoritmos Basados en Cúmulos de Partículas Para la Resolución de Problemas Complejos*.
- [18] García, J.M., Apolloni, J., Alba, E. and Leguizamón, G. *Algoritmo Basado en Cúmulos de Partículas y Evolución Diferencial para la Resolución de Problemas de Optimización Continua*. VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'09).
- [19] García, M. and Blaconá, M. (2003). *Métodos de Estimación para Datos Longitudinales con Información Faltante*. Instituto de Investigaciones Teóricas y Aplicadas de la Escuela de Estadística, Universidad Nacional de Rosario.
- [20] Glover, F. and Kochenberger, G. (2002). *Handbook of Metaheuristics*. Kluwer Academic Publishers, Norwell, MA.

- [21] Goldstein, H. (2003). *Multilevel Statistical Models*. Third Edition. London, Edward Arnold.
- [22] Gonul, F. and Srinivasan, K. (1993). *Modeling multiple sources of heterogeneity in multinomial logit models: Methodological and managerial issues*. Marketing Science, vol.12, n°. 3, verano, págs. 213-229.
- [23] Hachuel, L., Boggio, G. and Wojdyla, D. (2004). *Modelos Logit Mixtos: una aplicación en el área de la salud*. Instituto de Investigaciones Teóricas y Aplicadas de la Escuela de Estadística, Universidad Nacional de Rosario.
- [24] Hedeker, D. and Gibbons, R.D.(1996). *MIXOR: a computer program for mixed-effects ordinal probit and logistic regression analysis*. Computer Methods and Programs in Biomedicine, 49, 157-176.
- [25] Hesketh, S.R. and Skrondal, A.(2002). *Reliable Estimation Of Generalized Linear Mixed Models Using Adaptive Quadrature*. The Stata Journal 2, 1-21.
- [26] Hvass, M.E. (2010). *Good Parameter for Particle Swarm Optimization*. Hvass Laboratories Technical Report no. HL1001.
- [27] Hvass, M.E. (2010). *Good Parameters for Differential Evolution*. Hvass Laboratories Technical Report no. HL1002.
- [28] Jiang, J.(2007). *Linear and Generalized Linear Mixed Models and Their Applications*. Springer, New York, USA.
- [29] Johnson, M.S. (2007). *Marginal Maximum Likelihood Estimation of Item Response Models in R*. Journal Estatistical Software 20.
- [30] Kachman, S.D. *An Introduction to Generalized Linear Mixed Models*. Department of Biometry, University of Nebraska-Lincoln, 59-73.
- [31] Kennedy, J. (1997) *The Particle Swarm: Social Adaptation of Knowledge*. IEEE International Conference on Evolutionary Computation, pages 303-308.

- [32] Laird, N.M. and Ware, J.H. (1982). *Random-Effects Models for Longitudinal Data*. Biometrics, Vol. 38, No. 4, pp. 963-974.
- [33] Larsen, K., Petersen, J.H., Jorgensen, E.B. and Endahl, L. (2000). *Interpreting Parameters in the Logistic Regression Model with Random Effects*. Biometrics 56, 909-914.
- [34] McCullagh, P. and Nelder, J.A. (1989). *Generalized Linear Models*. (Second Edition). New York: Chapman and Hall.
- [35] Nash, S.G. and Sofer, A. (1990). *Assessing a search direction within a truncated-Newton method*. Operations Research Letters, 9, pp. 219-221.
- [36] Pant, M., Thangaraj, R. and Abraham, A. (2009). *DE-PSO: a new hybrid meta-heuristic for solving global optimization problems*. New Mathematics and Natural Computation, Accepted for publication.
- [37] Pierce, D.A. and Schafer, D.W. (1986). *Residuals in generalized linear models*. J. Am. Statist. Assoc 81, 977-86.
- [38] Price, K.V., Storn, R. and Lampinen, J. (2005). *Differential Evolution: A practical Approach to Global Optimization*. Springer-Verlag, London, UK.
- [39] Revelt, D. and Train, K. (1998). *Mixed Logit with Repeated Choices: Households' Choice of Appliance Efficiency Level*. Review of Economics and Statistics, Vol. LXXX, No. 4, pp. 647-657.
- [40] Stiratelli, R., Lair, N. and Ware, J.H. (1984). *Random-Effects Models for Serial Observations with Binary Response*. Biometrics 40, 961-971.
- [41] Storn, R. and Price, V. (1995). *Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*. Tech. Rep., TR-95012-ICSI.

- [42] Strenio, J.L.F., Weisberg, H.I. and Bryk, A.S. (1983). *Empirical bayes estimation of individual growth curve parameters and their relationship to covariates*. Biometrics, 39, 71-86.
- [43] Thangaraj, R., Pant, M.,Bouvry, P. and Abraham, A. (2011). *Particle swarm optimization: Hybridization perspectives and experimental illustrations*. Applied Mathematics and Computation, Journal homepage: www.elsevier.com/locate/amc.
- [44] Verbeke, G. and Molenberghs, G. (2000). *Linear Mixed Models for Longitudinal Data*. New York, Springer-Verlag.
- [45] Verbeke, G., Spiessens, B. and Lesaffre, E. (2001). *Condicionnal Linear Mixed Models*. The American Statistician 55, 25-34.
- [46] Wang, P. and Puterman, M.L. (1998). *Mixed Logistic Regression Models*. Journal of Agricultural, Biological, and Environmental Statistics 3, 175-200.
- [47] Zeger, S.L. and Karim, M.R. (1991). *Generalized linear models with random effects: A Gibbs sampling approach*. JASA, 86: 79-86.

Apéndice A

Pruebas

Teorema A.1. Si α_i , $i = 1, 2, \dots, n$, es una variable aleatoria tal que

$$\exp(\alpha_i) \sim \text{gamma}(k, \lambda)$$

entonces:

1. La función de distribución de α_i está dada por

$$h(\alpha_i|k, \lambda) = \frac{1}{\lambda^k \Gamma(k)} \exp(k\alpha_i) \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right)$$

2. $E[\alpha_i] = \psi(k) + \ln(\lambda)$

3. $\text{Var}(\alpha_i) = \psi'(k)$

donde:

$$\Gamma(k) = \int_0^{\infty} t^{k-1} \exp(-t) dt$$

$$\psi(k) = \frac{\partial}{\partial k} \ln(\Gamma(k)) = \frac{\Gamma'(k)}{\Gamma(k)}$$

$$\Gamma'(k) = \int_0^{\infty} t^{k-1} \exp(-t) \ln t dt$$

$$\psi'(k) = \frac{1}{\Gamma^2(k)} \Gamma'(k) \Gamma''(k) - (\Gamma'(k))^2$$

Demostración. 1. Sea

$$x = \exp(\alpha_i) \quad (\text{A.1})$$

como $x \sim \text{gamma}(k, \lambda)$, entonces la función de distribución de x es

$$f(x|k, \lambda) = \frac{1}{\lambda^k \Gamma(k)} x^{k-1} \exp\left(-\frac{x}{\lambda}\right) \quad 0 < x < \infty, \quad k > 0, \quad \lambda > 0 \quad (\text{A.2})$$

Ahora, de (A.1) podemos despejar α_i así

$$\alpha_i = \ln(x)$$

donde $\ln(x)$ es una función monótona creciente para $0 < x < \infty$.

Por otro lado, sabemos que la función $x = \exp(\alpha_i)$ es continua y que su derivada $x' = \exp(\alpha_i)$ también es continua, así que la función de la transformación $\alpha_i = \ln(x)$ se obtiene como

$$\begin{aligned} h(\alpha_i|k, \lambda) &= f(\exp(\alpha_i)) \left| \frac{d(\exp(\alpha_i))}{d\alpha_i} \right| \\ &= \frac{1}{\lambda^k \Gamma(k)} (\exp(\alpha_i))^{k-1} \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right) |\exp(\alpha_i)| \\ &= \frac{1}{\lambda^k \Gamma(k)} \exp(k\alpha_i) \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right) \end{aligned}$$

2.

$$\begin{aligned} E[\alpha_i] &= \int_0^\infty \alpha_i h(\alpha_i|k, \lambda) d\alpha_i \\ &= \int_0^\infty \alpha_i \frac{1}{\lambda^k \Gamma(k)} \exp(k\alpha_i) \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right) d\alpha_i \\ &= \frac{1}{\Gamma(k)} \int_0^\infty \left(\frac{\exp(\alpha_i)}{\lambda}\right)^k \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right) \alpha_i d\alpha_i \end{aligned}$$

Hacemos el cambio de variable

$$t = \frac{\exp(\alpha_i)}{\lambda} \quad \text{con} \quad dt = \frac{\exp(\alpha_i)}{\lambda} d\alpha_i = t d\alpha_i$$

Luego

$$\alpha_i = \ln(\lambda t) \quad \text{con} \quad d\alpha_i = \frac{dt}{t}$$

Por lo tanto, podemos escribir la esperanza de α_i como

$$\begin{aligned}
 E[\alpha_i] &= \frac{1}{\Gamma(k)} \int_0^\infty t^k \exp(-t) \ln(\lambda t) \frac{dt}{t} \\
 &= \frac{1}{\Gamma(k)} \left(\int_0^\infty t^{k-1} \exp(-t) \ln(t) dt + \int_0^\infty t^{k-1} \exp(-t) \ln(\lambda) dt \right) \\
 &= \frac{1}{\Gamma(k)} (\Gamma'(k) + \ln(\lambda)\Gamma(k)) \\
 &= \psi(k) + \ln(\lambda)
 \end{aligned}$$

3. Recordemos que $Var(\alpha_i) = E[\alpha_i^2] - (E[\alpha_i])^2$, así que encontremos primero $E[\alpha_i^2]$.

En efecto

$$\begin{aligned}
 E[\alpha_i^2] &= \int_0^\infty \alpha_i^2 h(\alpha_i|k, \lambda) d\alpha_i \\
 &= \int_0^\infty \alpha_i^2 \frac{1}{\lambda^k \Gamma(k)} \exp(k\alpha_i) \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right) d\alpha_i \\
 &= \frac{1}{\Gamma(k)} \int_0^\infty \left(\frac{\exp(\alpha_i)}{\lambda}\right)^k \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right) \alpha_i^2 d\alpha_i
 \end{aligned}$$

Hacemos el cambio de variable

$$t = \frac{\exp(\alpha_i)}{\lambda} \quad \text{con} \quad dt = \frac{\exp(\alpha_i)}{\lambda} d\alpha_i = t d\alpha_i$$

Luego

$$\alpha_i^2 = (\ln(\lambda t))^2 \quad \text{con} \quad d\alpha_i = \frac{dt}{t}$$

Por lo tanto la esperanza la escribimos como

$$\begin{aligned}
 E[\alpha_i^2] &= \frac{1}{\Gamma(k)} \int_0^\infty t^k \exp(-t) (\ln(\lambda t))^2 \frac{dt}{t} \\
 &= \frac{1}{\Gamma(k)} \left(\int_0^\infty t^{k-1} \exp(-t) \ln^2(t) dt + 2 \ln(\lambda) \int_0^\infty t^{k-1} \exp(-t) \ln(t) dt \right. \\
 &\quad \left. + \ln^2(\lambda) \int_0^\infty t^{k-1} \exp(-t) dt \right) \\
 &= \frac{1}{\Gamma(k)} (\Gamma''(k) + 2 \ln(\lambda)\Gamma'(k) + \ln^2(\lambda)\Gamma(k)) \\
 &= \frac{\Gamma''(k)}{\Gamma(k)} + 2 \ln(\lambda) \frac{\Gamma'(k)}{\Gamma(k)} + \ln^2(\lambda)
 \end{aligned}$$

Ahora podemos encontrar la varianza de α_i

$$\begin{aligned}
Var(\alpha_i) &= \frac{\Gamma''(k)}{\Gamma(k)} + 2\ln(\lambda)\frac{\Gamma'(k)}{\Gamma(k)} + \ln^2(\lambda) - \left(\frac{\Gamma'(k)}{\Gamma(k)} + \ln(\lambda)\right)^2 \\
&= \frac{\Gamma''(k)}{\Gamma(k)} - \frac{\Gamma'^2(k)}{\Gamma^2(k)} \\
&= \frac{1}{\Gamma^2(k)}\Gamma(k)\Gamma''(k) - \Gamma'^2(k) = \psi'(k)
\end{aligned}$$

□

Teorema A.2. Sea $Y_i = (y_{i1}, y_{i2}, \dots, y_{ij})$ el vector de J respuestas dicotómicas de un individuo i talque $p_{ij} = P(y_{ij} = 1|\alpha_i) = \exp(-\exp(\alpha_i + X_{ij}\beta))$. Si α_i es una variable aleatoria con distribución Log-gamma, entonces

$$E[y_{ij}] = \left(\frac{1/\lambda}{1/\lambda + \exp(X_{ij}\beta)}\right)^k$$

Demostración.

$$\begin{aligned}
E[y_{ij}] &= E[E(y_{ij}|\alpha_i)] \\
&= \int P(y_{ij} = 1|\alpha_i)h(\alpha_i)d\alpha_i \\
&= \int \exp(-\exp(\alpha_i + X_{ij}\beta)) \left(\frac{1}{\lambda^k\Gamma(k)} \exp(k\alpha_i) \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right)\right) d\alpha_i \\
&= \frac{1}{\lambda^k\Gamma(k)} \int \exp(k\alpha_i) \exp\left(-\exp(\alpha_i) \left(\exp(X_{ij}\beta) + \frac{1}{\lambda}\right)\right) d\alpha_i
\end{aligned}$$

Llamemos $\rho = \exp(X_{ij}\beta) + \frac{1}{\lambda}$ y multipliquemos por $\frac{\rho^k}{\rho^k}$ la integral anterior para escribirla como

$$= \frac{1}{\lambda^k\Gamma(k)} \int \frac{1}{\rho^k} (\rho \exp(\alpha_i))^k \exp(-\rho \exp(\alpha_i)) d\alpha_i$$

Ahora hacemos el cambio de variable $t = \rho \exp(\alpha_i)$ con $dt = \rho \exp(\alpha_i) d\alpha_i$ y $d\alpha_i = \frac{dt}{t}$, así tenemos

$$\begin{aligned}
E[y_{ij}] &= \frac{1}{\lambda^k\Gamma(k)} \frac{1}{\rho^k} \int t^k \exp(-t) \frac{dt}{t} \\
&= \frac{1}{\lambda^k} \frac{1}{\rho^k} \frac{1}{\Gamma(k)} \Gamma(k) \\
&= \frac{1}{\lambda^k} \frac{1}{\rho^k} \\
&= \frac{1}{\lambda^k} \left(\frac{1}{1/\lambda + \exp(X_{ij}\beta)}\right)^k
\end{aligned}$$

□

Teorema A.3. Sea $\mathbf{Y}_i = (y_{i1}, y_{i2}, \dots, y_{iJ})$ el vector de J respuestas dicotómicas e independientes de un individuo i talque $p_{ij} = P(y_{ij} = 1|\alpha_i) = \exp(-\exp(\alpha_i + X_{ij}\beta))$. Si α_i es una variable aleatoria con distribución Log - gamma(k, λ) entonces la función de verosimilitud marginal del individuo está dada por

$$L(\beta, k, \lambda) = \left(\frac{1/\lambda}{1/\lambda + \sum_{j=1}^J \exp(X_{ij}\beta)} \right)^k$$

Demostración. Como las respuestas del individuo son independientes entonces

$$P(\mathbf{Y}_i = 1|\alpha_i) = \prod_{j=1}^J P(y_{ij} = 1|\alpha_i) = \prod_{j=1}^J p_{ij}$$

Luego la función de verosimilitud marginal del individuo es

$$\begin{aligned} L(\beta, k, \lambda) &= \int P(\mathbf{Y}_i = 1|\alpha_i)h(\alpha_i)d\alpha_i \\ &= \int \prod_{j=1}^J p_{ij}h(\alpha_i)d\alpha_i \\ &= \int \prod_{j=1}^J \exp(-\exp(\alpha_i + X_{ij}\beta))h(\alpha_i)d\alpha_i \\ &= \int \exp\left(-\sum_{j=1}^J \exp(\alpha_i + X_{ij}\beta)\right)h(\alpha_i)d\alpha_i \\ &= \int \exp\left[-\exp(\alpha_i) \left(\sum_{j=1}^J \exp(X_{ij}\beta)\right)\right] \left[\frac{1}{\lambda^k \Gamma(k)} \exp(k\alpha_i) \exp\left(-\frac{1}{\lambda} \exp(\alpha_i)\right)\right] d\alpha_i \\ &= \frac{1}{\lambda^k \Gamma(k)} \int \exp\left[-\exp(\alpha_i) \sum_{j=1}^J \exp(X_{ij}\beta) + k\alpha_i - \frac{1}{\lambda} \exp(\alpha_i)\right] d\alpha_i \\ &= \frac{1}{\lambda^k \Gamma(k)} \int \exp(k\alpha_i) \exp\left[-\exp(\alpha_i) \left(\sum_{j=1}^J \exp(X_{ij}\beta) + \frac{1}{\lambda}\right)\right] d\alpha_i \end{aligned}$$

Llamemos $\rho = \sum_{j=1}^J \exp(X_{ij}\beta) + \frac{1}{\lambda}$ y multipliquemos por $\frac{\rho^k}{\rho^k}$ la integral anterior para escribirla como

$$L(\beta, k, \lambda) = \frac{1}{\lambda^k \Gamma(k)} \int \frac{1}{\rho^k} (\rho \exp(\alpha_i))^k \exp(-\rho \exp(\alpha_i)) d\alpha_i$$

Ahora hacemos el cambio de variable $t = \rho \exp(\alpha_i)$ con $dt = \rho \exp(\alpha_i) d\alpha_i$ y $d\alpha_i = \frac{dt}{t}$, así tenemos

$$\begin{aligned}
 L(\beta, k, \lambda) &= \frac{1}{\lambda^k \Gamma(k)} \frac{1}{\rho^k} \int t^k \exp(-t) \frac{dt}{t} \\
 &= \frac{1}{\lambda^k} \frac{1}{\rho^k} \frac{1}{\Gamma(k)} \int t^{k-1} \exp(-t) dt \\
 &= \frac{1}{\lambda^k} \frac{1}{\rho^k} \frac{1}{\Gamma(k)} \Gamma(k) \\
 &= \frac{1}{\lambda^k} \frac{1}{\rho^k} \\
 &= \left(\frac{1/\lambda}{1/\lambda + \sum_{j=1}^J \exp(X_{ij}\beta)} \right)^k
 \end{aligned}$$

□

Apéndice B

Programas en R

1. Algoritmo propuesto para la estimación usando Metaheurística

```
#Inicialización

N=50 #N: Número de individuos
m=15 #m: medidas repetidas por individuo

ealfa=matrix(rgamma(N, 50, scale = 1/2),nrow=N,ncol=1)
alfa=log(ealfa)
beta.=matrix(c(-4.8, 2.3),nrow=2,ncol=1)

#Matriz X_ cada fila corresponde al valor de xís de los individuos
#N filas x m columnas
X_<-matrix(rbinom(N*m,1,0.5),nrow=N,ncol=m)

#Vector de errores transpuesto
e<-matrix(rnorm(N*m,0,0.12),nrow=N,ncol=m)

#Vector de respuestas y probabilidades transpuesto
y<-matrix(NA,nrow=N,ncol=m)
p<-matrix(NA,nrow=N,ncol=m)

for (i in 1:N){
  for (j in 1:m){
    p[i,j]= exp(-exp(alfa[i]+ c(1,X_[i,j])%*%beta.+e[i,j]))
    y[i,j]<-rbinom(1,1,p[i,j])
  }
}

p.i <-matrix(NA,nrow=N,ncol=1)
for(i in 1:N){
  pro<-1
```

```

for(j in 1:m){
pro<- pro*p[i,j]
}
p.i[i]=pro
}

p.est<-matrix(NA,nrow=N,ncol=m)
p.ie <-matrix(NA,nrow=N,ncol=1)
error <-matrix(NA,nrow=N,ncol=1)

MSE<-function(a){
Ko=a[1]
Lo=a[2]
Bo=a[3:4]

ealfa.e=matrix(rgamma(N, Ko, scale = 1/Lo),nrow=N,ncol=1)
alfa.e=log(ealfa.e)
e.e<-e
#e.e<-matrix(rnorm(N*m,0,0.12),nrow=N,ncol=m)

for (i in 1:N){
for (j in 1:m){
p.est[i,j]= exp(-exp(alfa.e[i]+ c(1,X_[i,j])%*%Bo+e.e[i,j]))
}
}

for(i in 1:N){
pro<-1
for(j in 1:m){
pro<- pro*p.est[i,j]
}
p.ie[i]=pro

error[i]<-((p.i[i]-p.ie[i])*100)^2
}

return(sum(error))
}

PSO.ED<-function(N,p,min.,max.,Miter,f, CR, F.,vmax,psi1,psi2,w,inercia)
{
min.<-matrix(min.,nrow=1,ncol=p)
max.<-matrix(max.,nrow=1,ncol=p)

```

```

#Población inicial aleatoria
x<-matrix(c(rep(NA,p*N)),nrow=N,ncol=p)
for (l in 1:p){
x[,l]<- matrix(c(runif(N,min=min.[l],max=max.[l])), nrow=N,ncol=1)
}

#Matriz de velocidades iniciales
v0<-matrix(c(runif(N*p,min=-vmax,max=vmax)),nrow=N,ncol=p)

#Fitness de la población inicial
fi<-matrix(NA,nrow=N,ncol=1)
for (i in 1:N){
fi[i]<-f(x[i,])
}

#matriz que contiene los valores de la población inicial y el fitness
xi<-cbind(x,fi)

#mejor posición global
fgbest<-min(xi[, (p+1)])
xgbest<-xi[which.min(xi[, (p+1)]),1:p]

#mejor posición del individuo
xbesti<-x
fbesti<-fi

#Actualización el valor de la posición sumándole la velocidad hallada
for ( i in 1:N){
for (j in 1:p){
x[i,j]<-x[i,j]+v0[i,j]
}
}

#Inicialización
iter<-0
histo<-c(iter,xgbest,fgbest)

D.<-matrix(NA,nrow=1,ncol=p)
T.<-matrix(NA,nrow=1,ncol=p)
u<-matrix(NA,nrow=1,ncol=p)
y<-matrix(c(rep(NA,p*N)),nrow=N,ncol=p)

while(iter<Miter){
for (i in 1:N){

```

```

#Mutación
P<-c(1:N)
enteros<-sample(P[P!=i],3)
a<-enteros[1]
b<-enteros[2]
c.<-enteros[3]

T.<-x[a,]+F.*(x[b,]-x[c.,])

#Recombinación
for(j in 1:p){
if(runif(1,0,1)<CR){u[j]<-T.[j] }
else{u[j]<-x[i,j]}
}

#Verificación de las restricciones de borde
for (l in 1:p){
if(u[l]<min.[l]){u[l]<-min.[l]}
else if(u[l]>max.[l]){u[l]<-max.[l]}
else{u[l]<-u[l]}
}

#Selección
if(f(u)<f(x[i,])){
x[i,]<- u
}

else if (f(u)>=f(x[i,])){

if(inercia==1){
# Velocidad con factor de inercia
v0[i,]<-w*v0[i,]+psi1*runif(1,0,1)*(xbesti[i,]-x[i,])
+psi2*runif(1,0,1)*(xgbest-x[i,])
}

if(inercia==0){
# Velocidad con factor de constricción
psi<-psi1+psi2
k<-2/(abs(2-psi-sqrt(psi^2-4*psi)))
v0[i,]<-k*(v0[i,]+psi1*runif(1,0,1)*(xbesti[i,]-x[i,])
+psi2*runif(1,0,1)*(xgbest-x[i,]))
}

y[i,]<-x[i,]+v0[i,]

```

```

#Verificación de las restricciones de borde
for(j in 1:p){
  if(y[i,j]<min.[j]){y[i,j]<-min.[j]}
  else if(y[i,j]>max.[j]){y[i,j]<-max.[j]}
  else{y[i,j]<-y[i,j]}
}

  if(f(y[i,])<f(x[i,])){x[i,]<-y[i,]}
  else{x[i,]<-x[i,]}
}
}

#calculo del fitness de la nueva posición
for (i in 1:N){
fi[i]<-f(x[i,] )

if (fi[i]<fbesti[i])
{
      xbesti[i,]<-x[i,]
      fbesti[i]<-fi[i]
}
}

#Redefinición de la matriz xi
xi<-cbind(x,fi)

#Calculo del nuevo mejor punto y fitness
fgbest<-min(xi[, (p+1)])
xgbest<-xi[which.min(xi[, (p+1)]), 1:p]

iter<-iter+1

# Registro del resultado de cada iteración
histo<-rbind(histo,c(iter,xgbest,fgbest))
}

return(histo)
}

resumen<-function(x){
media<-mean(x)
desv.std<-sqrt(var(x))
coef.var<-(desv.std/media)*100
mediana<-median(x)
minimo<-min(x)

```

```

maximo<-max(x)
resul<-rbind(c('media','desv.std','coef.var','mediana','minimo','maximo'),
c(media,desv.std,coef.var,mediana,minimo,maximo))
return(resul)
}

desem<-function(n,N,Miter,inercia,f,CR, F.,psi1,psi2,w){
histo<-PSO.ED(N,4,c(48,1,-6.8,0.3),c(52,5,2.8,4.3),Miter,f,CR, F., 1.2,
psi1,psi2,w,inercia)
run<-1
while(run<n){
z<-PSO.ED(N,4,c(48,1,-6.8,0.3),c(52,5,2.8,4.3),Miter,f,CR, F., 1.2,
psi1,psi2,w,inercia)
histo<-rbind(histo,z)
run<-run+1
}
return(histo)
}

x1<-desem(10,17,50,1,MSE,0.7122,0.6301,-0.7238,2.0289,-0.3593)

```

2. Algoritmo de Estimación usando el paquete *glmmPQL*

```
#base de datos
```

```
N=50 #N: Número de individuos
m=15 #m: medidas repetidas por individuo
```

```
ealfa=matrix(rgamma(N, 50, scale = 1/2),nrow=N,ncol=1)
alfa=log(ealfa)
beta.=matrix(c(-4.8, 2.3),nrow=2,ncol=1)
```

```
#Matriz X_ cada fila corresponde al valor de xís de los individuos
#N filas x m columnas
X_<-matrix(rbinom(N*m,1,0.5),nrow=N,ncol=m)
```

```
#Vector de errores transpuesto
e<-matrix(rnorm(N*m,0,0.12),nrow=N,ncol=m)
```

```
#Vector de respuestas y probabilidades transpuesto
y<-matrix(NA,nrow=N,ncol=m)
p<-matrix(NA,nrow=N,ncol=m)
```

```

for (i in 1:N){
for (j in 1:m){
  p[i,j]= exp(-exp(alfa[i]+ c(1,X_[i,j])%%beta.+e[i,j]))
  y[i,j]<-rbinom(1,1,p[i,j])
}
}

#Base de datos para modelado con cuadratura gaussiana

id <- factor(rep(1:50, rep(15, 50)))
y.<-matrix(t(y),nrow=15*50,ncol=1)
x <-matrix(t(X_),nrow=15*50,ncol=1)
dat <- data.frame(id = id, y = y., x = x)

#-----
library(MASS)

model<-glmmPQL(y ~ x, random = ~ 1 | id, family = binomial, data = dat)
summary(model)

```