



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**Estimación de parámetros de Ecuaciones Diferenciales
Ordinarias con algoritmos genéticos e híbridos,
con aplicaciones**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
MATEMÁTICO**

P R E S E N T A:

LOUIS DAVID BRETON TENORIO



**DIRECTOR DE TESIS:
DR. JESÚS LÓPEZ ESTRADA**

2014



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Hoja de Datos del jurado

1. Datos del alumno

Breton

Tenorio

Louis David

63947688

Universidad Nacional Autónoma de México

Facultad de Ciencias

Matemáticas

410022683

2. Datos del tutor

Dr

Jesús

López

Estrada

3. Datos del sinodal 1

Dr

Pedro Eduardo

Miramontes

Vidal

4. Datos del sinodal 2

M en C

Guillermo

Gómez

Alcaraz

5. Datos del sinodal 3

M en C

María Lourdes

Velasco

ArreguÍ

6. Datos del sinodal 4

Dra

María de Lourdes

Esteva

Peralta

7. Datos del trabajo escrito.

Estimación de parámetros de Ecuaciones Diferenciales Ordinarias
con algoritmos genéticos e híbridos, con aplicaciones.

Estimación de parámetros de EDO's

62p

2014

Índice general

Resumen	1
1. Una introducción a la estimación de parámetros	3
1.1. ¿Qué es la estimación de parámetros?	3
1.2. La estimación de parámetros como un problema de optimización	5
1.2.1. Introducción	5
1.2.2. Funciones residuales	5
1.2.3. Funciones objetivo	6
1.2.4. El parámetro estimado	7
1.3. Estimación de parámetros por mínimos cuadrados	7
1.3.1. Motivación	7
1.3.2. Elementos básicos	8
1.3.3. El caso lineal	9
1.3.4. Métodos iterativos para el caso no lineal	10
1.3.4.1. Esquema de los métodos iterativos	11
1.3.4.2. Método de Cauchy para máximo descenso	12
1.3.4.3. Método de Newton	12
1.3.4.4. Método de Gauss	13
1.3.4.5. Método de Levenberg-Marquardt	14
1.4. Un vistazo a los intervalos de confianza	15
2. Algoritmos evolutivos	17
2.1. ¿Qué son los algoritmos evolutivos?	17
2.1.1. Trasfondo biológico	17
2.1.2. Esquema general de los AE	18
2.2. Elementos de los Algoritmos evolutivos	19
2.2.1. Representación del individuo	19
2.2.2. La función del fitness (o función evaluadora)	19
2.2.3. La población	19
2.2.4. El proceso de selección de los padres	20
2.2.5. Los operadores de recombinación y mutación	20
2.2.5.1. Mutación	20
2.2.5.2. Recombinación	20
2.2.5.3. El mecanismo de selección de los sobrevivientes	20

2.3.	Ejemplos de algoritmos genéticos para la optimización de funciones	21
2.3.1.	Representación de los individuos	21
2.3.1.1.	Representación binaria	21
2.3.1.2.	Representación por números reales	23
2.3.2.	Ejemplo de mutaciones para la representación binaria	24
2.3.3.	Mutaciones para la representación por números reales	25
2.3.3.1.	Mutación utilizando variables aleatorias uniformes	25
2.3.3.2.	Mutación utilizando variables aleatorias normales	26
2.3.4.	Recombinación	26
2.3.4.1.	Recombinaciones para la representación binaria	26
2.3.4.2.	Recombinaciones para la representación por números reales	27
2.3.4.3.	Recombinación aritmética total	27
2.3.4.4.	Recombinación aritmética simple	28
2.3.5.	Mecanismos de selección de Padres	28
2.3.5.1.	Fitness proporcional	28
2.3.6.	Mecanismos de selección de supervivientes	29
2.3.6.1.	Selección fitness proporcional (μ, λ) :	29
2.3.6.2.	Selección fitness proporcional $(\mu + \lambda)$:	30
2.3.6.3.	Selección con elitismo	30
2.4.	Estrategias evolutivas para la optimización de funciones	30
2.4.1.	Representación	30
2.4.2.	Mutación	31
2.4.2.1.	Mutación con sólo un parámetro estratégico	31
2.4.2.2.	Mutación con n parámetros estratégicos	32
2.4.3.	Recombinación	32
2.4.3.1.	Recombinación discreta	32
2.4.3.2.	Recombinación intermedia	33
2.4.4.	Mecanismos de selección de Padres	33
2.4.5.	Mecanismos de selección de supervivientes	34
2.4.5.1.	Mecanismo de selección (μ, λ)	34
2.4.5.2.	Mecanismo de selección $(\mu + \lambda)$	34
3.	Estimación de Parámetros en EDO's con aplicaciones	35
3.1.	El problema de valores iniciales para EDOs	35
3.1.1.	Formulación del problema (PVI) para EDOs	35
3.1.2.	Dos teoremas fundamentales para el estudio de parámetros de EDOs	35
3.2.	El problema de estimación de parámetros en EDOs	36
3.2.1.	Métodos de estimación de parámetros en EDOs	37
3.2.1.1.	Diferenciación de los datos	37
3.2.1.2.	Integración de los datos	38
3.2.1.3.	Integración de la ecuación diferencial	40

3.2.2.	Ejemplo de aplicación de los métodos	41
3.2.2.1.	Diferenciación de los datos:	41
3.2.2.2.	Integración de los datos:	41
3.2.2.3.	Integración de la ecuación diferencial:	42
3.3.	Programa de estimación de parámetros para EDO's	43
3.3.1.	Algoritmos y métodos utilizados	43
3.3.2.	Detalles técnicos	44
3.3.2.1.	Lenguaje de programación	44
3.3.2.2.	Bibliotecas usadas	44
3.3.3.	Sitio del programa	44
3.3.3.1.	Links de los tutoriales del programa	44
3.3.3.2.	Links de descarga del programa	45
3.3.4.	Vista previa del programa	45
3.4.	Aplicaciones de la estimación de parámetros de EDO's.	46
3.4.1.	Estimación de parámetros artificiales	46
3.4.1.1.	Sistema Lotka-Volterra	46
3.4.1.2.	Modelo SIR	48
3.4.2.	Estimación de parámetros con datos reales	51
3.4.2.1.	Población de México, modelo con la logística	51
3.4.2.2.	El caso de la obesidad en México con SIR modificado	53

Agradecimientos **59**

Bibliografía **61**

Resumen

Usualmente en la Ciencia se recopilan tablas de datos de un fenómeno en particular, en los cuales los científicos tratan de modelar con funciones matemáticas. A estas funciones se le llaman el modelo matemático del fenómeno y suelen tener parámetros con interpretación física, química, biológica, económica, etc.

Justamente la estimación de parámetros es el proceso en el cual se trata de determinar estos coeficientes (o parámetros) de tal manera que el modelo matemático se ajuste lo mejor posible a los datos recopilados, y de esta forma poder efectuar un análisis de validez o rechazo al modelo planteado.

Resulta que con frecuencia en la investigación y las aplicaciones la función matemática está dada de manera implícita. Por ejemplo, como la solución de un problema de ecuaciones diferenciales ordinarias (EDOs) con valores iniciales, pero con parámetros en estas ecuaciones desconocidos.

Este trabajo tiene como objetivo explicar la teoría que hay detrás de la estimación de parámetros, y la creación de un software que aplique esta misma para así poder estimar los parámetros de modelos de EDO con datos reales y simulados.

En una primera instancia describiremos lo que es la estimación no lineal de parámetros, y algunos de los métodos usuales que se utilizan para realizar esta tarea (por ejemplo, el método de Levenberg-Marquardt y el de Newton o máximo descenso).

En una segunda instancia presentaremos una clase de algoritmos llamados algoritmos genéticos, en particular su teoría matemática y su aplicación en la estimación no lineal de parámetros.

Por último, presentaremos la parte central de este proyecto el cual consiste en la estimación no lineal de parámetros en modelos de EDOs con el desarrollo de software propio escrito en C++, utilizando algoritmos genéticos y algoritmos híbridos ("combinaciones de los métodos usuales con algoritmos genéticos"). Además, se presentará un manual de usuario del software.

1 Una introducción a la estimación de parámetros

Referencias del capítulo: [ALR],[BRD],[DTT],[KSB],[MRS]

En este primer capítulo se presentará el problema de estimación de parámetros, se explicará cómo se puede plantear este problema como uno de optimización. Por último, daremos a conocer la teoría ya desarrollada para resolver un caso de este tipo de problemas (optimización de mínimos cuadrados).

1.1. ¿Qué es la estimación de parámetros?

Usualmente en la ciencia se recopilan tablas de datos de un fenómeno en particular, en los cuales los científicos tratan de modelar con funciones matemáticas. A estas funciones se les llaman el modelo matemático del fenómeno y suelen tener parámetros que tienen interpretación física. De un punto de vista determinista, si el modelo es correcto existen los parámetros exactos que describen al fenómeno estudiado. Sin embargo, los datos recopilados tienen un grado de precisión limitado, por lo cual es imposible obtener exactamente los verdaderos valores de los parámetros.

Justamente la estimación de parámetros es el proceso en el cual se trata de determinar estos coeficientes (o parámetros) de tal manera que el modelo matemático se ajuste lo mejor posible a los datos recopilados, y de esta forma dar validez o rechazo al modelo planteado.

Este problema aparece principalmente en la física, la biología y la química. Algunos ejemplos de aplicaciones clásicos son (ver capítulo 3, sección 3.3):

1. El cálculo de órbitas. La órbita de un satélite puede ser expresada como una función del tiempo, en la cual sus parámetros describen la fuerza de atracción que ejercen los objetos masivos cercanos a él. Estos parámetros pueden ser determinados observando la órbita de dicho satélite.
2. En la exploración geofísica. Teniendo varias mediciones de la intensidad del campo magnético de una región, se pueden detectar anomalías, las cuales pueden indicar depósitos de oro.

3. En la epidemiología. Tomando medidas, a lo largo del tiempo, de una población infectada por un virus se pueden deducir los parámetros del modelo S.I.R. (susceptibles, infectados y recuperados).
4. En biología. Tomando medidas de presa-depredador de poblaciones se pueden estimar los parámetros de Lotka-Volterra.

Un ejemplo muy sencillo, para ilustrar, es encontrar la recta que mejor aproxima los siguientes puntos:

$$P1 = (0, 0), P2 = (2, 2).$$

Dado que la ecuación de la recta está dada por:

$$y = ax + b$$

Lo que tenemos que hacer es encontrar los parámetros “a” y “b” que mejor se ajusten a estos datos. Lo cual, en este caso, no resulta nada complicado puesto que se puede plantear un sistema de ecuaciones para encontrar tales parámetros (i.e $a = 1$ y $b = 0$). Pero el problema se complica más si tenemos 3 puntos, puede ser que éstos no sean colineales y por lo tanto no será tan sencillo encontrar la mejor recta.

Para extrapolarnos de nuestro ejemplo supongamos que tenemos un fenómeno que queremos estudiar y para ello se recopiló un conjunto de datos $D \subset \mathbb{R}^k \times \mathbb{R}^r$, gracias a ciertos instrumentos de medición. Una forma explícita de D estaría dado por:

$$D := \{(x_i, y_i)\}_{i \in 1, 2, \dots, n}$$

Con $x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^r, \forall i \in 1, 2, \dots, n$.

Ahora, también supongamos que tenemos un modelo matemático, el cual depende de variables de estado y de parámetros, el cual describe comportamientos del objeto de estudio. Justamente se piensa que ajustando los parámetros se debería poder reproducir los datos obtenidos y de esta manera capturar información del fenómeno que viene codificado en los parámetros. Formalizando un poco consideramos al modelo como una función que estaría dada por:

$$f(x, \theta) = (f_1(x, \theta), f_2(x, \theta), \dots, f_r(x, \theta))^T \quad (1.1)$$

Donde, $f : \mathbb{R}^k \times \Theta \mapsto \mathbb{R}^r$, representa una función r – *dimensional* que modela al fenómeno y $\Theta \subset \mathbb{R}^p$.

$f_i : \mathbb{R}^{k \times p} \mapsto \mathbb{R}$, son las componentes de la función f .

$\theta = (\theta_1, \theta_2, \dots, \theta_p)^T \in \mathbb{R}^p$ representa el vector p – *dimensional* de los parámetros del modelo.

$x = (x_1, x_2, \dots, x_k)^T \in \mathbb{R}^k$ representa el vector k – *dimensional* de las variables de estado.

La estimación de parámetros consiste en encontrar $\hat{\theta}$ de tal manera que:

$$f(x_i, \hat{\theta}) \simeq y_i \forall i \in \{1, 2, 3, \dots, n\}$$

1.2. La estimación de parámetros como un problema de optimización

1.2.1. Introducción

Resulta bastante natural pensar este problema como un problema de optimización, puesto que nuestro objetivo es ajustar lo mejor posible el modelo a nuestro datos, lo cual sencillamente resulta en minimizar la distancia entre nuestros datos y nuestros puntos generados por el modelo.

Tomando a nuestro conjunto de datos de la forma:

$$D := \{(x_i, y_i)\}_{i \in 1, 2, \dots, n}$$

Con $x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^r, \forall i \in 1, 2, \dots, n$; y nuestro modelo como (1.1), dado por:

$$f(x, \theta) = (f_1(x, \theta), f_2(x, \theta), \dots, f_r(x, \theta))^T$$

De manera determinista vamos a suponer que **existen los parámetros reales del fenómeno estudiado**, es decir existe θ^* tal que:

$$f_j(x_i, \theta^*) = y_{ij} + \epsilon_{ij} \tag{1.2}$$

Donde y_{ij} es la proyección de la j – *ésima* coordenada del vector y_i (el cual se denotará como i_j – *ésimo* dato), y ϵ_{ij} corresponde al error de medición en el i_j – *ésimo* dato.

1.2.2. Funciones residuales

Las funciones residuales son funciones que tienen la forma de:

$$e_{ij}(\theta) = |y_{ij} - f_j(x_i, \theta)|$$

Donde $\|\cdot\|_2$ es la norma euclidiana e y_{ij} es la proyección de la j – *ésima* coordenada del vector y_i .

Estas funciones residuales nos dan la distancia entre un dato (por esto mismo se les llama residuales) y nuestra aproximación obtenida en ese punto con el parámetro θ . Se sigue naturalmente de (1.2) que:

$$e_{ij}(\theta^*) = |\epsilon_{ij}|$$

1.2.3. Funciones objetivo

Ahora que tenemos todas las herramientas definiremos nuestra función objetivo (o a veces llamada función error), la cual se denotará por ϕ :

$$\phi(\theta) = \sum_{i=1}^n \sum_{j=1}^r w_{ij} [e_{ij}(\theta)]^p \quad (1.3)$$

Donde $w_{ij} \in \mathbb{R}^+$ y $p \in \mathbb{N}$.

Los escalares w_{ij} en (1.3) se les denominan pesos y, como su nombre lo dice, se encargan de maximizar o minimizar la importancia de ciertos datos en el proceso de estimación de parámetros, también su objetivo puede ser el escalamiento de datos de distintas unidades.

En un caso hipotético, **supongamos que el i_k -ésimo dato es mucho más preciso que el j_k -ésimo dato**. Lo que más nos interesa, claramente, es que nuestro modelo se ajuste más al i_k -ésimo que al j_k -ésimo dato, en otras palabras queremos penalizar más a nuestra función objetivo si tiene valores lejanos al i_k -ésimo que cuando tiene valores lejanos al j_k -ésimo.

De manera formal si tenemos:

$$|\epsilon_{ik}| < |\epsilon_{jk}|$$

Se tomaría a w_i y w_j de manera que:

$$w_{ik} > w_{jk} > 0$$

De esta forma le damos más importancia al i_k -ésimo dato que al j_k -ésimo.

Si $w_{ij} \neq 1$ a la función ϕ se le llama función objetivo con pesos, y viceversa si $w_{ij} = 1$ a la función ϕ se le llama función objetivo sin pesos.

Observación. Un ejemplo típico de una función objetivo sin pesos es tomar en (1.3) a $w_{ij}(x) = 1$ y $p = 2$, es decir, nuestra función objetivo sería la suma de los cuadrados de los errores.

$$\phi(\theta) = \sum_{i=1}^n \sum_{j=1}^r e_{ij}(\theta)^2 = \sum_{i=1}^n |y_{ij} - f_j(x_i, \theta)|^2 \quad (1.4)$$

El proceso por el cuál se minimiza esta función se le denomina mínimos cuadrados (o Min Q's). La estudiaremos a detalle en la siguiente sección.

1.2.4. El parámetro estimado

Justamente la estimación de parámetros como un problema de optimización puede ser vista como el proceso por el cual se obtiene:

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \phi(\theta) \quad (1.5)$$

Donde $\hat{\theta}$ se le denomina **parámetro estimado**.

Recordando que θ^* (1.2) es el **parámetro exacto**, nos queda la pregunta si se cumple que:

$$\hat{\theta} \simeq \theta^* \quad (1.6)$$

Este tema en particular se llama **identificación de parámetros**, el cuál no abordaremos en esta tesis, pero analiza la veracidad de la ecuación (1.6).

Una respuesta parcial es: si Θ (espacio de los parámetros) es convexo y ϕ es convexa en Θ , entonces se tiene que $\hat{\theta}$ existe y es único. Por lo que,

$$|\phi(\theta^*) - \phi(\hat{\theta})| \leq \sum_{i=1}^n \sum_{j=1}^r w_{ij} |\epsilon_{ik}|^p + \phi(\hat{\theta}) \quad (1.7)$$

Donde ϵ_{ij} es el error de medición en el i_j -ésimo dato.

Suponiendo que la parte derecha de (1.7) es bastante pequeña. Se puede hacer un **análisis de intervalos de confianza** para estudiar precisamente (1.6) (ver sección 1.4).

1.3. Estimación de parámetros por mínimos cuadrados

1.3.1. Motivación

Sin pérdida de generalidad, recordando la ecuación (1.3) de la sección anterior podemos definir a nuestra función objetivo como:

$$\phi(\theta) = \sum_{i=1}^n \sum_{j=1}^r e_{ij}(\theta)^p = \sum_{i=1}^n \|y_i - f(x_i, \theta)\|_2^p$$

Los métodos que veremos a continuación utilizan información sobre la derivada de ϕ , en particular si $p = 1$ la norma euclidiana no es diferenciable, y por lo tanto no será posible utilizar la derivada de ϕ .

Si $p = 2$ se tiene que $\|\cdot\|_2^2$ es diferenciable, además $\|\cdot\|_2^2$ puede ser asociada con el producto interior euclidiano, esto nos da una ventaja al tener la posibilidad de definir el concepto de ortogonalidad.

Sin embargo, si $p \neq 2$ y $p > 1$ perdemos el concepto de ortogonalidad, lo cual nos dificulta mucho la tarea de optimización, especialmente cuando f es lineal en θ .

Por esto mismo, se decide tomar a la función objetivo como:

$$\phi(\theta) = \sum_{i=1}^n \|y_i - f(x_i, \theta)\|_2^2$$

1.3.2. Elementos básicos

A continuación, los vectores se tomarán como vectores columnas, dada una función $\phi : \mathbb{R}^p \rightarrow \mathbb{R}$, la derivada parcial de $\phi(\theta)$ con respecto a su i -ésima coordenada (si existe) se representará como:

$$\frac{\partial \phi}{\partial \theta_i}$$

El gradiente de una función $\phi : \mathbb{R}^p \rightarrow \mathbb{R}$ estará dado por (cuando exista):

$$\nabla \phi = \left(\frac{\partial \phi}{\partial \theta_1}, \frac{\partial \phi}{\partial \theta_2}, \dots, \frac{\partial \phi}{\partial \theta_l} \right)^T$$

El hessiano de un función $\phi : \mathbb{R}^p \rightarrow \mathbb{R}$ evaluado en $\theta \in \mathbb{R}^p$ se entenderá como una matriz $H \in M_{p \times p}(\mathbb{R})$ donde:

$$H_{ij} = \frac{\partial^2 \phi}{\partial \theta_i \partial \theta_j}$$

Tomaremos a nuestro conjunto de datos de la forma:

$$D := \{(x_i, y_i)\}_{i \in \{1, 2, \dots, n\}}$$

Con $x_i \in \mathbb{R}^k, y_i \in \mathbb{R}^r, \forall i \in \{1, 2, \dots, n\}$

Y nuestro modelo como en (1.1):

$$f(x, \theta)$$

$f : \mathbb{R}^k \times \Theta \mapsto \mathbb{R}^r$ representa una función r -dimensional que modela al fenómeno y $\Theta \subset \mathbb{R}^p$.

$\theta = (\theta_1, \theta_2, \dots, \theta_p)^T \in \mathbb{R}^p$ representa el vector p – *dimensional* de los parámetros del modelo.

$x = (x_1, x_2, \dots, x_k)^T \in \mathbb{R}^k$ representa el vector k – *dimensional* de las variables de estado.

Se toma a nuestra función error como la suma de los errores al cuadrado:

$$\phi(\theta) = \sum_{i=1}^n \|y_i - f(x_i, \theta)\|_2^2$$

Supondremos que **existen los parámetros reales del fenómeno estudiado**, es decir, existe θ^* tal que:

$$f_j(x_i, \theta^*) = y_{ij} + \epsilon_{ij}$$

Donde éstas ϵ_i corresponden al error de medición en el i_j – *ésimo* dato.

Por otro lado, llamaremos a $\hat{\theta}$ como el vector de parámetros estimado, el cual corresponde a:

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \phi(\theta)$$

1.3.3. El caso lineal

Supongamos que el modelo es lineal sobre los parámetros, es decir, que nuestra función f puede ser representada como:

$$f(x, \theta) = \sum_{i=1}^p \theta_i f_i(x)$$

Donde $f : \mathbb{R}^{k \times p} \mapsto \mathbb{R}^r$, $f_i : \mathbb{R}^k \mapsto \mathbb{R}^r$, $\theta = (\theta_1, \theta_2, \dots, \theta_p)^T \in \mathbb{R}^p$.

Resulta ser que si nuestro modelo es lineal sobre θ , este problema se vuelve un problema de proyección ortogonal. Para poder apreciar esto vemos a nuestra función de errores al cuadrado es decir:

$$\begin{aligned} \phi(\theta) &= \sum_{i=1}^n \|y_i - f(x_i, \theta)\|^2 \\ &= \sum_{i=1}^n \|y_i - \sum_{i=1}^p \theta_i f_i(x_i)\|^2 \end{aligned}$$

Esta función también se puede ver como:

$$\phi(\theta) = \|Y - X\theta\|^2 \tag{1.8}$$

Donde $Y = (y_1, y_2, \dots, y_n)^T$, y $X = \begin{pmatrix} f_1(x_1) & f_2(x_1) & \dots & f_p(x_1) \\ f_1(x_2) & f_2(x_2) & \dots & f_p(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n) & f_2(x_n) & \dots & f_p(x_n) \end{pmatrix}$ (notemos que X es una matriz de $n \times p$, con bloques de tamaño $1 \times r$).

Resulta que, geoméricamente, se puede interpretar como la proyección del vector Y en el subespacio generado por X . Por lo tanto se tiene que:

$$X^T X \theta = X^T Y \quad (1.9)$$

Las cuales se denominan ecuaciones normales. Si nuestra matriz X es de rango máximo entonces se sigue que $X^T X$ es no singular y, por lo tanto, podemos asegurar que la solución existe y es única.

$$\theta = (X^T X)^{-1} X^T Y \quad \text{si } X \text{ tiene rango máximo}$$

Observación. Como una pequeña observación, numéricamente este problema no se resuelve directamente con las ecuaciones normales puesto que puede estar mal condicionado. En lugar de eso, se emplea un método utilizando descomposición QR de la matriz X .

1.3.4. Métodos iterativos para el caso no lineal

Nuestro objetivo, antes dicho, es minimizar la función objetivo $\phi : \mathbb{R}^p \rightarrow \mathbb{R}$, la cual en este caso **no es lineal**, para ello, presentaremos a continuación una gama de métodos que se basan en un **proceso iterativo**. Primero seleccionamos un vector de parámetros iniciales, θ_1 , gracias a éste generaremos una sucesión $\theta_2, \theta_3, \dots$, la cuál esperamos que converja a $\hat{\theta}$. El procedimiento por el cual obtenemos a θ_{i+1} a partir de θ_i se le denomina iteración. Entre cada iteración hay un paso, el cual denominaremos s_i y puede ser expresado, naturalmente, como:

$$s_i = \theta_{i+1} - \theta_i$$

Como lo que queremos es alcanzar un mínimo, es natural pedir que:

$$\phi(\theta_{i+1}) < \phi(\theta_i) \quad (1.10)$$

Llamaremos a la ecuación (1.10) la condición de aceptabilidad, esta misma nos indica si el paso s_i es aceptable o no.

A los pasos s_i los representaremos de la siguiente manera:

$$s_i = \lambda_i d_i \tag{1.11}$$

Donde $d_i \in \mathbb{R}^p$ representa la dirección en la que nos desplazamos y donde λ_i es un escalar.

En este contexto a d_i se le denomina **dirección del paso**, y a λ_i como **tamaño del paso**.

Nuestro objetivo ahora es encontrar un condición para que nuestros pasos s_i sean aceptables. Para ello, introduciremos una función auxiliar ψ_i dada por:

$$\psi_i(p) = \phi(\theta_i + \lambda d)$$

Esta función ψ_i es un función de \mathbb{R} a \mathbb{R} y si su derivada negativa significa que ésta misma es decreciente. Es decir, que si ψ_i' es negativa, podemos garantizar que cuando empezamos a movernos desde θ_i en la dirección de d la función ϕ decrece. Y por lo tanto, tomando un λ lo suficientemente pequeño podemos garantizar que $s = \lambda d$ es un paso aceptable. De aquí mismo diremos que si $\psi_i' < 0$ entonces d es una dirección aceptable.

Gracias a esto podemos formular el siguiente teorema:

Teorema. *Una dirección d es aceptable si, y sólo si, existe una única matriz positivamente definida R tal que:*

$$d = -R[\nabla\phi(\theta_i)]$$

Donde $\nabla\phi = (\frac{\partial\phi}{\partial\theta_1}, \frac{\partial\phi}{\partial\theta_2}, \dots, \frac{\partial\phi}{\partial\theta_i})^T$ es el gradiente de ϕ .

La demostración de este teorema se la dejaremos al lector, puede consultarse en[BRD].

Si nos preguntamos sobre la **convergencia** de la sucesión, recordemos que si nuestro modelo es correcto entonces suponemos la existencia de θ^* mínimo de ϕ . Esto significa que la sucesión $\{\phi(\theta_i)\}$ **es acotada por abajo y puesto que es monótona podemos garantizar que tiene al menos un punto de convergencia**. En la práctica, sin embargo, el radio de convergencia puede ser tan bajo que la sucesión $\{\phi(\theta_i)\}$ puede dar la apariencia de que no converge.

1.3.4.1. Esquema de los métodos iterativos

Los métodos que presentaremos a continuación tienen todos el siguiente esquema:

1. Para $i = 1$ se toma un vector parámetro θ_1 (se le denomina parámetro inicial).
2. Se determina el vector d_i de la ecuación (1.4) de manera que sea un dirección aceptable.
3. Se determina el tamaño de paso λ_i de manera que s_i cumpla la condición (1.3).
4. Se verifica algún criterio de término. Si éste no se satisface se regresa al paso 2.

Observación. Tomando $\epsilon > 0$ bastante pequeño, una condición de convergencia sería:

$$\phi(\theta_j) < \epsilon$$

1.3.4.2. Método de Cauchy para máximo descenso

Para el método de máximo descenso de Cauchy (1848) simplemente se toma, en el esquema iterativo anterior, la dirección:

$$d_i = -\nabla\phi(\theta_i)$$

Al menos inicialmente ésta es la dirección dónde la función objetivo $\phi(\theta)$ tiene su máximo descenso, de ahí el nombre del método.

Como paso se toma a $s_i = \lambda_i d_i$ con $\lambda_i = 1$. Por lo tanto, el proceso iterativo viene dado por:

$$\theta_{i+1} = \theta_i + d_i = \theta_i - \nabla\phi(\theta_i)$$

La ventaja de este método es que es bastante simple, desafortunadamente este método requiere de demasiados pasos para su convergencia, tienden a zigzaguear. Este método difícilmente se utiliza en aplicaciones prácticas y sólo se menciona por referencia histórica.

Observación. Se necesita que $\phi \in C^1(\Theta)$ para garantizar la existencia de $\nabla\phi$ en Θ .

1.3.4.3. Método de Newton

Para explicar este método consideremos la expansión de Taylor de segundo orden de ϕ alrededor de θ_i . Es decir:

$$Q_i(\theta) = \phi(\theta_i) + [\nabla\phi(\theta_i)]^T(\theta - \theta_i) + \frac{1}{2}(\theta - \theta_i)^T H(\theta_i)(\theta - \theta_i)$$

Donde H es el hessiano de nuestra función objetivo (recuérdese del cálculo multivariable ([MRS])) que Q_i es la mejor aproximación cuadrática de ϕ en θ_i .

Como nuestro objetivo es encontrar un mínimo, es natural buscar un punto para el cual el gradiente de Q_i se anule. Es decir, θ_{i+1} tal que:

$$\nabla Q_i(\theta_{i+1}) = \nabla \phi(\theta_i) + H(\theta_i)(\theta_{i+1} - \theta_i) = 0 \quad (1.12)$$

Si $H(\theta_i)$ es positivamente definido (esto implica que $\det(H(\theta_i)) \neq 0$) la ecuación (1.12) tiene solución única θ_{i+1} , la cual está dada por:

$$\theta_{i+1} = \theta_i - H^{-1}(\theta_i)[\nabla \phi(\theta_i)], \quad i = 1, 2, \dots \quad (1.13)$$

La ecuación (1.13) define el proceso iterativo del método de Newton. Se toma, en el esquema principal, como dirección:

$$d_i = -H^{-1}(\theta_i)[(\nabla \phi)(\theta_i)]$$

Con tamaño de paso $\lambda_i = 1$.

La belleza del método de Newton radica en su rápida convergencia (cuadrática) local cuando la condición inicial θ_1 se encuentra en una vecindad cercana al mínimo θ^* de ϕ . Su desventaja es que se requiere del cálculo de segundas derivadas para obtener el hessiano H , lo cual numéricamente suele ser costoso.

Observación. Para este método se toma $\phi \in C^2(\Theta)$ para garantizar la existencia de $\nabla \phi$ y H en Θ .

1.3.4.4. Método de Gauss

Carl F. Gauss (1777-1855) desarrolló el método de mínimos cuadrados, con el objetivo de determinar las trayectorias elípticas de los planetas y otros cuerpos del sistema solar, utilizando puntos observados sobre sus trayectorias.

Una manera sencilla de determinar este método es tomando a:

$$F(\theta) = (f(x_1, \theta), f(x_2, \theta), \dots, f(x_n, \theta))^T$$

Podemos ver a ϕ como:

$$\phi(\theta) = \frac{1}{2} \|Y - F(\theta)\|^2$$

Donde $Y = (y_1, y_2, \dots, y_n)^T$.

Haciendo una aproximación de Taylor a primer orden en θ_i de $F(\theta)$ obtenemos que:

$$\phi(\theta) \cong \frac{1}{2} \|Y - F(\theta_i) - \nabla F(\theta_i)^T(\theta - \theta_i)\|^2$$

Utilizando las ecuaciones (1.8) y (1.9) del caso lineal obtenemos:

$$\nabla F(\theta_i) \nabla F(\theta_i)^T (\theta_{i+1} - \theta_i) = \nabla F(\theta_i) (Y - F(\theta_i)) \quad (1.14)$$

Si $\nabla F(\theta_i) \nabla F(\theta_i)^T$ es definida positivamente podemos retomar el esquema principal de esta sección y utilizando la ecuación (1.14) se deduce que la dirección de paso viene dada por:

$$d_i = \left(\nabla F(\theta_i) \nabla F(\theta_i)^T \right)^{-1} [\nabla F(\theta_i) (Y - F(\theta_i))] \quad (1.15)$$

Con un tamaño de paso $\lambda_i = 1$. Recordando que $-\nabla \phi(\theta_i) = [\nabla F(\theta_i) (Y - F(\theta_i))]$, la ecuación (1.15) puede ser reescrita como:

$$d_i = - \left(\nabla F(\theta_i) \nabla F(\theta_i)^T \right)^{-1} \nabla \phi(\theta_i)$$

La inconveniencia de este método nace si el gradiente $\nabla F(\theta_i)$ es cercanamente singular. Resulta que la longitud del paso s_i puede llegar a ser muy grande.

Observación. Se toma $F \in C^1(\Theta)$ para garantizar la existencia de ∇F y $\nabla \phi$ en Θ .

1.3.4.5. Método de Levenberg-Marquardt

Levenberg (1944) y Marquardt (1963) propusieron un algoritmo para resolver el problema del método de Gauss, el cual consiste en introducir una componente $\gamma_i B$ en la dirección de descenso d_i , esta dirección esta dada por:

$$d_i = - \left(\nabla F(\theta_i) \nabla F(\theta_i)^T + \gamma_i B \right)^{-1} \nabla \phi(\theta_i)$$

Donde $\gamma_i \in \mathbb{R}, \gamma_i \geq 0$ y B es una matriz definida positivamente de dimensiones $p \times p$.

En el método original de Levenberg-Marquardt se toma a la matriz B como la matriz identidad I de dimensiones $p \times p$. Lo cual implica que la dirección de descenso es:

$$d_i = - \left(\nabla F(\theta_i) \nabla F(\theta_i)^T + \gamma_i I \right)^{-1} \nabla \phi(\theta_i) \quad (1.16)$$

Observemos que el método es un combinación del método de máximo descenso y el método de Gauss.

Si $\gamma_i = 0$ la ecuación (1.16) se transforma en:

$$d_i = - \left(\nabla F(\theta_i) \nabla F(\theta_i)^T \right)^{-1} \nabla \phi(\theta_i)$$

La cual corresponde a la dirección del método de Gauss.

Por otro lado, si $\gamma_i > 0$ podemos reescribir (1.9):

$$d_i = - \left(\gamma_i \left[\frac{1}{\gamma_i} \nabla F(\theta_i) \nabla F(\theta_i)^T + I \right] \right)^{-1} \nabla \phi(\theta_i) \quad (1.17)$$

Tomando $\lim_{\gamma_i \rightarrow +\infty}$ de la ecuación (1.17) obtenemos que la dirección de descenso es:

$$d_i = -\nabla \phi(\theta_i)$$

La cuál es la dirección de máximo descenso de Cauchy.

Para la selección de las γ_i se toma el siguiente algoritmo:

1. Cuando $i = 1$ tomemos a γ_i como un número pequeño (ej: $\lambda = 0.1$).
2. En la iteración i -ésima se calcula $\phi(\theta_i + d_i)$ donde d_i está dado por (1.9).
3. Si $\phi(\theta_i + v_i) < \phi(\theta_i)$, se toma a $\theta_{i+1} = \phi(\theta_i + v_i)$ y se toma a $\gamma_{i+1} = \max(0.1\lambda, \epsilon)$. Donde ϵ es un número muy pequeño (ej $\epsilon = 10^{-8}$). De otra forma pasamos al paso 4.
4. Se toma un tamaño de paso s_i de manera que $\phi(\theta_i + s_i d_i) < \phi(\theta_i)$, por lo que $\theta_{i+1} = \phi(\theta_i + s_i d_i)$. Por último, se toma a $\lambda_{i+1} = 10\lambda_i$ y se regresa al paso 2.

Se ejecutan sistemáticamente los pasos anteriores hasta que se cumpla una condición de término.

Observación. Al igual que en el método de Gauss se toma $F \in C^1(\Theta)$ para garantizar la existencia de ∇F y $\nabla \phi$ en Θ .

1.4. Un vistazo a los intervalos de confianza

Bajo el supuesto de que $\hat{\theta}$ (el parámetro estimado) existe y es único analizaremos la siguiente desigualdad:

$$|\phi(\theta^*) - \phi(\hat{\theta})| < \epsilon \quad (1.18)$$

donde $\epsilon \in \mathbb{R}^+$.

Para analizar (1.19) analizaremos el siguiente conjunto:

$$A = \{\theta \in \Theta \mid |\phi(\theta) - \phi(\hat{\theta})| < \epsilon\}$$

Observación. De la ecuación (1.7) si $\epsilon = \sum_{i=1}^n \sum_{j=1}^r w_{ij} |\epsilon_{ik}|^2 + \phi(\hat{\theta})$ se tiene que $\theta^* \in A$.

Supongamos que $\phi \in C^2(\Theta)$, recordando de cálculo multivariable, la serie de Taylor de segundo orden al rededor de $\hat{\theta}$ es:

$$\phi(\theta) \simeq \phi(\hat{\theta}) + [\nabla \phi(\hat{\theta})]^T (\theta - \hat{\theta}) + \frac{1}{2} (\theta - \hat{\theta})^T H(\hat{\theta}) (\theta - \hat{\theta})$$

Como $\hat{\theta} = \arg \min_{\theta \in \Theta} \phi(\theta)$ tenemos que $\nabla \phi(\hat{\theta}) = 0$ por lo tanto, de (1.8) tenemos:

$$|(\theta - \hat{\theta})^T H(\hat{\theta})(\theta - \hat{\theta})| < 2\epsilon \quad (1.19)$$

La ecuación (1.20) nos define un elipsoide llamado la ϵ -*region de indiferencia*. Esta región resulta ser un elipsoide y el análisis de éste mismo nos determina que tanto puede variar $\hat{\theta}$ de θ^* .

2 Algoritmos evolutivos

Referencias del capítulo: [BCK],[ESM].[GNT]

Los algoritmos evolutivos (AEs), son métodos de optimización o de búsqueda de soluciones inspirados en la teoría de la evolución de Charles Darwin, de ahí mismo el nombre de “evolutivos”.

Para la optimización de funciones reales, es importante resaltar que este tipo de algoritmos son ajenos de los típicos métodos de cálculo, en particular **no se requiere para nada la derivada** de la función. Además, bajo ciertos supuestos se sabe que tienen probabilidad 1 de converger al óptimo (ver [GNT]).

En este capítulo daremos a conocer en abstracto las componentes de estos algoritmos. Y por último daremos ejemplos de éstos para la optimización de funciones reales, los cuáles utilizaremos para el programa de estimación de parámetros que se presentará en esta tesis.

2.1. ¿Qué son los algoritmos evolutivos?

2.1.1. Trasfondo biológico

En el libro "The Darwinian theory of evolution", se explica la adaptabilidad de las especies a su entorno por el principio de la selección natural, es decir, la presión del medio ambiente sólo permite que a lo largo del tiempo los individuos más aptos sobrevivan.

Pero también otro factor importante reconocido por Darwin es la aparición de pequeñas variaciones, aparentemente aleatorias, en los fenotipos (características físicas) de los padres y de su descendencia. Se sabe que gracias a la bioquímica y la genética se confirmó y extendió la teoría de Darwin. Por lo regular a ésta se le denomina neodarwinismo y se basa en la teoría de los genes.

Los genes pueden considerarse como la unidad primordial para explicar la herencia y la mutación, el conjunto de los genes (el total de la información genética) de un individuo se denomina genotipo y éste al expresarse en el medio ambiente constituye el fenotipo. Se sabe que parte de los genes de un progenitor (padres) se heredan a su descendencia, y que a su vez, en éste proceso suelen ocurrir pequeñas mutaciones en los genes, las cuáles pueden, o no, mejorar la adaptabilidad del individuo.

Los AEs se inspiran precisamente en este último proceso, y por esto se toma prestado vocabulario biológico en sus definiciones.

2.1.2. Esquema general de los AE

Existen muchas variantes de estos algoritmos, sin embargo, la idea detrás de todos éstos es la misma. Dada una población inicial de individuos (candidatos de soluciones), la presión del medio ambiente (función objetivo) fuerza una selección natural, lo que resulta en un aumento de "fitness" (adaptabilidad) de la población.

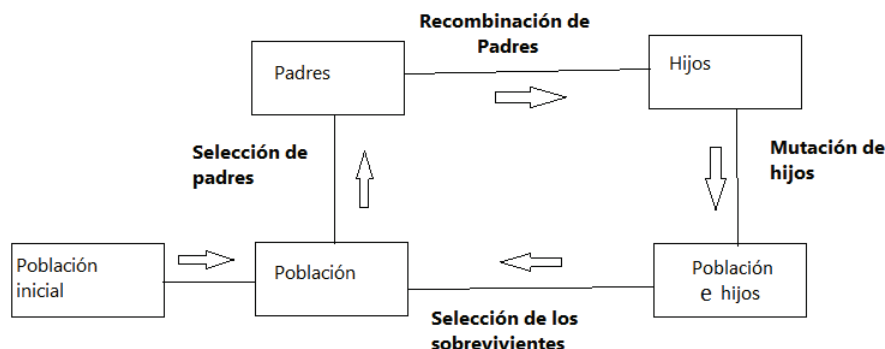
Es decir, dada una función objetivo que se desea maximizar (o minimizar), se crean un conjunto aleatorio de candidatos en el dominio de la función, a la cuál se le denomina población cero (o generación cero).

Se evalúa el fitness (adaptabilidad) de cada individuo en la población con la función objetivo como un abstracto del fitness (calidad). Basados en éste, se selecciona a los candidatos (determinísticamente o probabilísticamente) para que generen la siguiente población (o generación) utilizando operadores de recombinación y de mutación.

La **recombinación es aplicada a dos o más individuos** seleccionados, a los cuáles se les **conoce como padres**, y como resultado de está combinación **tenemos uno o más individuos llamados hijos**.

Por otro lado, la **mutación se le aplica a una cierta cantidad de hijos** y de esta manera (recombinando y mutando) creamos nuevos individuos que a su vez serán evaluados y competirán por un lugar en la población. El proceso se repite hasta que tengamos un candidato con un fitness bastante alto o hasta que una condición de término sea alcanzada (por ejemplo un número máximo de generaciones).

El siguiente diagrama ejemplifica los algoritmos:



2.2. Elementos de los Algoritmos evolutivos

Los algoritmos evolutivos tienen un cierto número de componentes, y operadores los cuáles al ser especificados determinan un algoritmo evolutivo (AE) en particular. Los componentes más importantes son:

2.2.1. Representación del individuo

La representación de los individuos es lo que nos permite ligar el problema matemático con los algoritmos evolutivos.

Como dicho anteriormente, en biología los genotipos son los conjuntos de genes que definen a un individuo, y los fenotipos son la expresión de esos genes en las características del individuo. En los AEs, se decide llamar **fenotipos a las posibles soluciones del problema** original y **genotipos al conjunto de datos que representa la información de estos fenotipos**.

Los individuos sólo contienen genotipos. El paradigma de la representación es decidir que genotipos se usarán para el problema (ejemplo: bits o números reales) y a su vez la función que asocia a los genotipos con los fenotipos.

2.2.2. La función del fitness (o función evaluadora)

La función del fitness, es la que **permite medir la calidad de un individuo**, y la que nos permite caracterizar el desempeño de nuestro algoritmo. Matemáticamente podemos representar la función fitness de la siguiente manera:

$$\phi : X \longrightarrow \mathbb{R}$$

Donde ϕ es la función del fitness, X es el espacio de los individuos.

Normalmente, se toma a la función ϕ definida positivamente y de manera que cuanto más alto sea el fitness de un individuo mejor es este mismo como solución del problema.

2.2.3. La población

La población es la parte central de los AE, **su papel es guardar las posibles soluciones. Es un conjunto finito y dinámico de individuos**, la población cambia y se adapta para mejorar (si es posible) el fitness de los individuos de la cuál está conformada. En la mayoría de los casos el número de individuos de la población es una constante.

2.2.4. El proceso de selección de los padres

El proceso de selección de los padres es la herramienta que permite a los individuos de mejor calidad ser progenitores (padres) de la siguiente generación. Los padres (o progenitores) son aquellos individuos que fueron seleccionados para pasar por los operadores de variación con el fin de crear un hijo para la siguiente generación.

Normalmente, este proceso de selección es aleatorio, pero tiene la particularidad de asociar una mayor probabilidad de selección a los individuos de mejor calidad e inversamente una probabilidad más baja a los individuos de menor calidad. El hecho de que también se puedan seleccionar individuos con un bajo fitness permite en lo general conservar la diversidad de la población y evitar quedar atrapado en un óptimo local.

2.2.5. Los operadores de recombinación y mutación

2.2.5.1. Mutación

La mutación es un operador que toma a un individuo y lo modifica azarosamente gracias a una cierta cantidad de variables aleatorias. Su finalidad es ayudar con la exploración del espacio de soluciones. En lo general se le asigna a las modificaciones pequeñas una mayor probabilidad e inversamente a las modificaciones grandes una menor. Gracias a que tenemos la probabilidad superior a cero de hacer un gran salto en el espacio de soluciones, entonces podemos evitar quedar atrapados en un óptimo local, lo cual resulta útil. Hay que notar que la mutación se sitúa al nivel de los genotipos, es por este motivo que las mutaciones dependen de la representación que fue seleccionada.

2.2.5.2. Recombinación

La recombinación es un operador que toma dos o más individuos para juntar sus características y de esta forma producir uno o más individuos. En principio, se mezclan a los individuos para juntar sus cualidades en un hijo, lo cual está directamente inspirado de la naturaleza. Al igual que el operador de mutación, se utilizan variables aleatorias para juntar los genotipos de los individuos seleccionados. Su finalidad es la exploración del espacio y la refinación de la solución.

2.2.5.3. El mecanismo de selección de los sobrevivientes

El mecanismo de selección de los sobrevivientes es muy similar al mecanismo de selección de los padres, su función es seleccionar los individuos que conformarán la siguiente generación, partiendo de la generación anterior y los hijos

generados por la selección de padres. Normalmente, también se selecciona azarosamente, pero dando prioridad a los individuos de mejor calidad. A este proceso también se le puede llamar **estrategia de reemplazamiento**.

2.3. Ejemplos de algoritmos genéticos para la optimización de funciones

En esta sección describiremos uno de los algoritmos evolutivos más conocidos. En particular nos concentramos en el problema de minimizar una función:

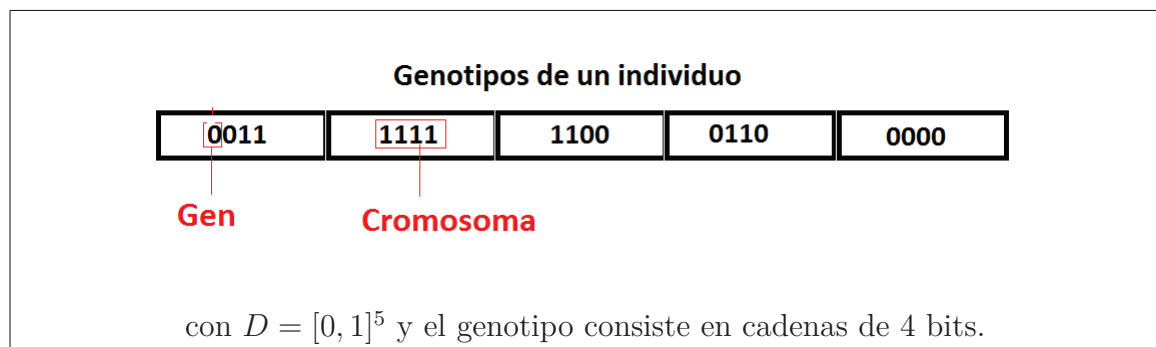
$$\phi : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}^+$$

Suponiendo que tenemos $D = [I_1, S_1] \times [I_2, S_2] \times \dots \times [I_n, S_n]$.

Presentaremos en una primera instancia dos tipos de representaciones para atacar este problema. Después, los distintos tipos de operadores variacionales (mutación y recombinación) para cada una de estas representaciones. Por último presentaremos los mecanismos de selección y reemplazamiento que utilizaremos para manejar la población de soluciones.

2.3.1. Representación de los individuos

2.3.1.1. Representación binaria



La representación binaria en este caso consiste en que los **genotipos de los individuos sean cadenas de bits**. Cada individuo tiene asociado un vector, en el cual cada entrada es una cadena de bits, este vector tiene la forma de:

$$V_{binario} = ((gen_{10}, gen_{11}, gen_{12}, \dots, gen_{1(m-1)}), (gen_{20}, gen_{21}, gen_{22}, \dots, gen_{2(m-1)}), \dots, (gen_{n0}, gen_{n1}, gen_{n2}, \dots, gen_{n.(m-1)}))$$

Donde $gen_{ij} \in \mathbb{Z}_2$ representa **un gen del genotipo**, m es el tamaño de la cadena de bits (el **tamaño de 1 cromosoma**), n es el número de entradas del vector binario (el **número de cromosomas**). Para simplificar llamemos a $C_i = (gen_{i0}, gen_{i1}, gen_{i2}, \dots, gen_{i(m-1)})$. (Nota C_i puede ser entendido como cromosoma) Por lo tanto, nuestro vector binario tiene la siguiente forma:

$$V_{binario} = (C_1, C_2, \dots, C_n) \in (\mathbb{Z}_2)^{m \times n}$$

Este vector representa los genotipos de nuestro individuo y, por lo tanto, tenemos que crear un función que asocie a éste mismo a los fenotipos, es decir, a un vector. Llamaremos T a esa función y por lo tanto tiene que cumplir:

$$\begin{aligned} T : (\mathbb{Z}_2)^{m \times n} &\longrightarrow D \\ T : (V) \in D \forall V &\in (\mathbb{Z}_2)^{m \times n} \end{aligned}$$

La forma más natural de hacer esto es con la ayuda del cambio de base de binario a enteros. Llamaremos BN a la **función que transforma un binario en un natural**. Por lo tanto:

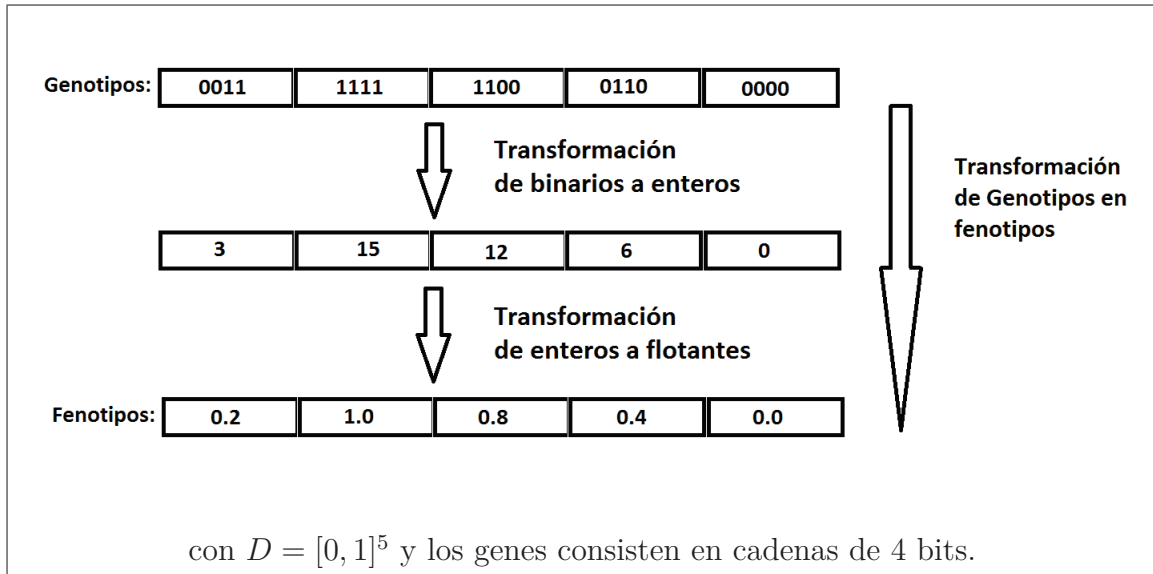
$$BN(b_0, b_1, \dots, b_{m-1}) = \sum_{i=0}^{m-1} 2^i * b_{(m-1)-i}$$

Ahora consideremos a L_i la función lineal afín que manda al intervalo $[0, 2^{m-1}]$ al intervalo $[I_i, S_i]$, de manera que $L_i(0) = I_i$ y $L_i(2^{m-1}) = S_i$. La expresión analítica de esta recta está dada, claramente, por:

$$L_i(x) = x * \frac{(S_i - I_i)}{2^{m-1}} + I_i$$

Por lo tanto, gracias a las funciones L_i y BN podemos dar una expresión de T , la cual sería:

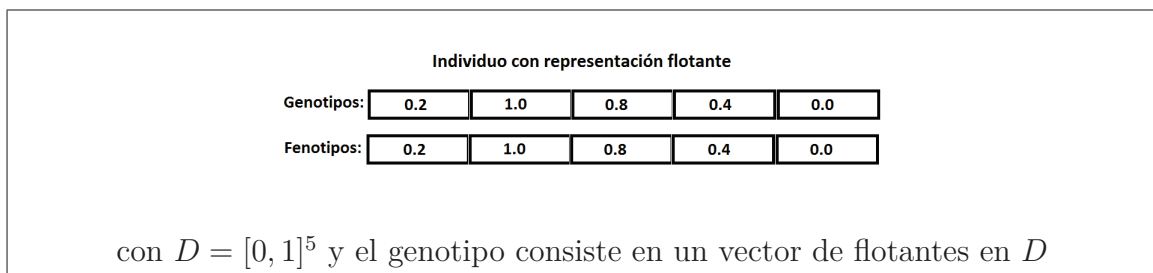
$$\begin{aligned} T(C_1, C_2, \dots, C_n) &= ((L_1 \circ BN)(C_1), \dots, (L_n \circ BN)(C_n)) \\ &= (L_1(BN(C_1)), \dots, L_n(BN(C_n))) \end{aligned}$$



En el fondo lo que hicimos es una discretización de cada intervalo $[I_i, S_i]$ en 2^{m-1} puntos. Es precisamente por esta razón que nuestra precisión, de nuestra solución, está acotada superiormente en cada entrada del vector, por:

$$Mprecision_i = \frac{(S_i - I_i)}{2^{m-1}}$$

2.3.1.2. Representación por números reales



Esta representación suele ser de las mejores para nuestro problema puesto que nuestro espacio es continuo y no discreto. La representación por números reales es bastante directa. Se toman a los genotipos de los individuos como:

$$\text{Genotipos} = (x_1, x_2, \dots, x_n)$$

donde $x_i \in [I_i, S_i]$ $i \in \{1, 2, \dots, n\}$.

En este caso los **genotipos coinciden con fenotipos** y, por lo tanto, no hay necesidad de definir una transformación T que asocie a éstos dos como en el caso anterior.

Claramente también su precisión está limitada por la precisión que pueden alcanzar los números flotantes de la máquina.

2.3.2. Ejemplo de mutaciones para la representación binaria

La mutación para representación binaria más usada es la de considerar a cada gen separadamente y hacer lo que se llama un **bit-flip**, dependiendo del resultado de una variable aleatoria, el gen 0 se transforma en 1 y el gen 1 en 0. (Ver Figura: Ejemplo de mutación)

Para describir esto matemáticamente introducimos el operador:

$$\begin{aligned} flip : \mathbb{Z}_2 &\longrightarrow \mathbb{Z}_2 \\ flip(gen) &= \begin{cases} gen, & \text{si } W \leq 1 - Pm \\ gen + 1, & \text{si } W < Pm \end{cases} \end{aligned}$$

donde W es un variable aleatoria uniforme en el $(0, 1)$, y $Pm \in [0, 1] \subset \mathbb{R}$ es la probabilidad de mutación, puesto que:

$$\begin{aligned} P(muta(gen) = gen + 1) &= P\{W < Pm\} = Pm \\ P(muta(gen) = gen) &= P\{W > Pm\} = 1 - Pm \end{aligned}$$

Gracias a este operador $flip$ construimos nuestro operador mutación, el cual está dado de la siguiente forma:

$$\begin{aligned} Mut : (\mathbb{Z}_2)^m &\longrightarrow (\mathbb{Z}_2)^m \\ Mut(gen_1, gen_2, \dots, gen_m) &= (muta(x_1), muta(x_2), \dots, muta(x_m)) \end{aligned}$$

Nótese que la probabilidad de $k < n$ bits (genes) cambien de estado está dada por la distribución binomial, i.e:

$$P(k \text{ bits cambien de estado}) = \binom{n}{k} Pm^k (1 - Pm)^{n-k}$$

Un ejemplo sencillo suponiendo que $m = 6$ es:

En este caso, 3 bits fueron cambiados

Cromosoma original

1	1	0	0	1	0
---	---	---	---	---	---

Mut

Cromosoma mutado

0	1	1	1	1	0
---	---	---	---	---	---

bits que resultaron cambiados por la mutación

Esta mutación podría corresponder a tomar $pm = 0.8$, y como resultado de las variables aleatorias:

Var. aleatoria W	0.7	0.9	0.5	0.3	0.85
Menor que pm	si	no	si	si	no
Mutación	si	no	si	si	no

En este caso, 3 bits fueron cambiados.

2.3.3. Mutaciones para la representación por números reales

La mutación para números reales es bastante sencilla de explicar matemáticamente. Lo que queremos hacer es un operador Mut de la siguiente forma:

$$Mut : D \rightarrow D$$

Para este operador se utilizan variables aleatorias continuas. A continuación, presentaremos posibles definiciones de este operador.

2.3.3.1. Mutación utilizando variables aleatorias uniformes

En este caso, queremos introducir una perturbación gracias a variables aleatorias uniformes de la siguiente forma:

$$Mut(x_1, x_2, \dots, x_n) = (x_1 + W_1(x_1), x_2 + W_2(x_2), \dots, x_n + W_n(x_n))$$

Donde $W_i(x)$ son variables aleatorias independientes que se distribuyen de la siguiente forma:

$$W_i(x) \sim U(I_i - x, S_i - x)$$

Claramente, se sigue de la definición que $W_i(x)$ si $x \in [I_i, S_i]$ entonces $x + W_i(x) \in [I_i, S_i]$. Y por lo tanto nuestro operador cumple que $Mut(D) \subseteq D$.

2.3.3.2. Mutación utilizando variables aleatorias normales

Al introducir una mutación utilizando variables aleatorias normales, se tiene que tomar en cuenta que al modificar una entrada con ésta no se salga del dominio de definición de nuestra función, por lo tanto, recurrimos a las siguientes funciones:

$$H_i : \mathbb{R} \longrightarrow [I_i, S_i]$$

$$H_i(x) = \begin{cases} x, & \text{si } I_i \leq x \leq S_i \\ I_i & \text{si } x < I_i \\ S_i, & \text{si } x > S_i \end{cases}$$

Y entonces, definamos nuestro operador *Mut* como:

$$Mut(x_1, x_2, \dots, x_n) = (H_1(x_1 + N_1), H_2(x_2 + N_2), \dots, H_n(x_n + N_n))$$

Donde $\{N_i\}_{i \in \{1, 2, \dots, n\}}$ son variables aleatorias que se distribuyen normalmente con esperanza 0 y varianza σ^2 (σ es definida según los casos, es decir, según cuánto se quiere perturbar, y su elección se deja al experimentador).

2.3.4. Recombinación

Al igual que la mutación, la recombinación se sitúa en los genotipos, precisamente por esto se tienen distintos tipos de recombinación, según la representación escogida.

2.3.4.1. Recombinaciones para la representación binaria

La recombinación que presentaremos a continuación es la más usada y se denomina one-point crossover. La idea básica detrás de esto es seccionar los cromosomas de los padres de manera aleatoria y recombinar estas secciones de cromosomas para crear nuevos hijos (esto está directamente inspirado en reproducción celular, es decir, la Meiosis) (Ver Subsubsección 2.3.4.1).

Introduzcamos primero el operador mezclar:

$$mezclar : (\mathbb{Z}_2)^m \times (\mathbb{Z}_2)^m \times \{2, \dots, m-1\} \longrightarrow (\mathbb{Z}_2)^m$$

$$mezclar(x, y, i) = (x_1, x_2, \dots, x_i, y_{i+1}, \dots, y_m)$$

Donde $x = (x_1, x_2, \dots, x_m) \in (\mathbb{Z}_2)^m$ y $y = (y_1, y_2, \dots, y_m) \in (\mathbb{Z}_2)^m$.

Gracias a este operador podemos definir nuestro operador estocástico de recombi-

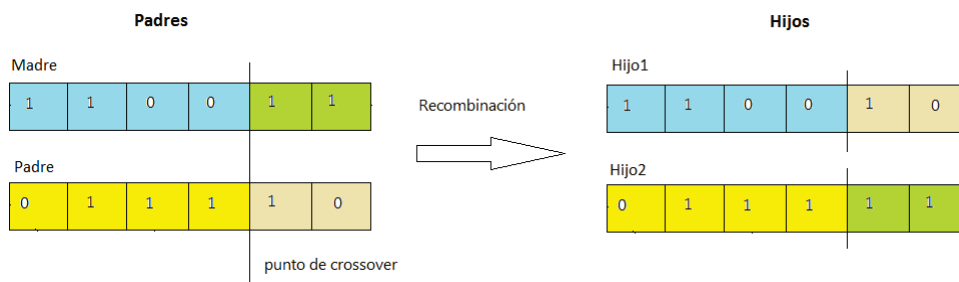
nación:

$$reco : (\mathbb{Z}_2)^m \times (\mathbb{Z}_2)^m \longrightarrow (\mathbb{Z}_2)^m \times (\mathbb{Z}_2)^m$$

$$reco(x, y) = \begin{cases} (x, y), & \text{si } Z \leq 1 - pr \\ (mezclar(x, y, W), mezclar(x, y, W)), & \text{si } Z < pr \end{cases}$$

Donde W es una variable aleatoria, uniforme y discreta en $\{2, 3, \dots, m - 1\}$, Z es una variable aleatoria y uniforme en el $(0, 1)$, y por último $pr \in [0, 1]$ se le conoce como la probabilidad de recombinación.

Un ejemplo de una recombinación sería, suponiendo que $m = 6$ (el tamaño de un cromosoma), $W = 4$ (resultado de la variable aleatoria uniforme discreta).



2.3.4.2. Recombinaciones para la representación por números reales

A continuación, mostraremos 2 tipos de recombinación para números reales las cuales son aritméticas, y dependen de un parámetro $\alpha \in [0, 1]$. La mayoría del tiempo se toma a $\alpha = \frac{1}{2}$, pero también se puede tomar como una variable aleatoria y uniforme en el $(0, 1)$.

2.3.4.3. Recombinación aritmética total

Esta recombinación es bastante simple, se basa en tomar un promedio aritmético de los padres. Esto puede ser descrito de la siguiente forma:

$$Reco : D \times D \longrightarrow D \times D$$

$$Reco(x, y) = (\alpha x + (1 - \alpha)y, \alpha y + (1 - \alpha)x)$$

Donde α es un constante en el intervalo $[0, 1]$. En este caso el hijo1 sería $\alpha x + (1 - \alpha)y$ y el hijo2 sería $\alpha y + (1 - \alpha)x$.

Observación. Este operador está bien definido en nuestro caso, puesto que claramente D es convexo.

2.3.4.4. Recombinación aritmética simple

La idea de esta recombinação también es bastante sencilla, se trata de tomar aleatoriamente las primeras k coordenadas del padre e insertarlas directamente en el hijo, las $n - k$ coordenadas faltantes se toman como el promedio aritmético de las últimas $n - k$ coordenadas de la madre y del padre. Definiremos el operador *mezclar* para ayudarnos a describir este procedimiento:

$$\text{Mezclar} : D \times D \times \{1, 2, \dots, m\} \longrightarrow D \times D$$

$$\text{Mezclar}(x, y, k) = ((x_1, \dots, x_{k-1}, \varepsilon_k, \dots, \varepsilon_n), (y_1, \dots, y_{k-1}, \eta_k, \dots, \eta_n))$$

con

$$\varepsilon_j = \alpha x_j + (1 - \alpha) y_j$$

$$\eta_j = \alpha y_j + (1 - \alpha) x_j$$

Donde $x \in D \subset \mathbb{R}^m$, $y \in D \subset \mathbb{R}^m$. Entonces *Reco* está dado por:

$$\text{Reco}(x, y) = \text{Mezclar}(x, y, W)$$

Donde W es una variable aleatoria, discreta y uniforme en $\{2, 3, \dots, m-1\}$ y $x, y \in D$.

2.3.5. Mecanismos de selección de Padres

2.3.5.1. Fitness proporcional

En general, como se describió en la sección 2.3, el mecanismo de selección tiene que cumplir que entre mejor sea el individuo más probabilidad tiene de ser seleccionado para ser progenitor (padre) de la siguiente generación. En este algoritmo el mecanismo de selección se le denomina **fitness proporcional**.

Para describir el siguiente método de selección supongamos que tenemos una población $Pob = (x_1, x_2, \dots, x_n)$, que consta de n individuos. Se define al fitness de cada individuo en Pob de la siguiente manera:

$$\text{Fitness}(x_i) = \phi(x_i)$$

El **fitness total** de una población se define como la suma de los fitness de cada individuo de la población, es decir, que este *fitness_total* es dado por:

$$\begin{aligned} \text{fitness_total}(Pob) &= \sum_{i=1}^n \text{Fitness}(x_i) \\ &= \sum_{i=1}^n \phi(x_i) \end{aligned}$$

Recordemos que nuestro objetivo es asignar una mayor probabilidad de selección a las mejores individuos, pero también al mismo tiempo queremos que todos los individuos tenga una probabilidad superior a 0 de ser seleccionados. En nuestro caso, dado que queremos minimizar la función ϕ , los mejores individuos son los que tienen un fitness más pequeño. Se sigue entonces que la manera natural de definir la probabilidad de que un individuo sea seleccionado corresponde a:

$$\begin{aligned} P\{x_i \text{ sea seleccionado}\} &= 1 - \frac{Fitness(x_i)}{fitness_total(Pob)} \\ &= 1 - \frac{\phi(x_i)}{\sum_{j=1}^n \phi(x_j)} \end{aligned} \quad (2.1)$$

Aunque ésta manera rompe un poco con el paradigma del fitness, lleva acabo su finalidad puesto que asignamos una mayor probabilidad de selecciones a los mejores individuos. Esto puede verse tomando $x_a, x_b \in Pob$ de forma que $\phi(x_a) < \phi(x_b)$:

$$\begin{aligned} \Rightarrow 1 - \frac{\phi(x_a)}{\sum_{i=1}^n \phi(x_i)} &> 1 - \frac{\phi(x_b)}{\sum_{i=1}^n \phi(x_i)} \\ \Rightarrow P\{x_a \text{ sea seleccionado}\} &> P\{x_b \text{ sea seleccionado}\} \end{aligned}$$

2.3.6. Mecanismos de selección de supervivientes

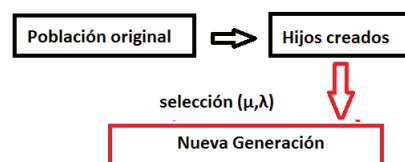
De ahora en adelante el número de individuos en la **población se representa por la constante μ** y el número de **hijos que son creados se representa por la constante λ** .

El mecanismo de selección de supervivientes es responsable de seleccionar los individuos μ que formarán la siguiente generación, como en la selección de los padres, se utiliza una selección de **fitness proporcional (ecuación 1.1)**.

Sin embargo, el conjunto en el cual se hace esta selección puede variar:

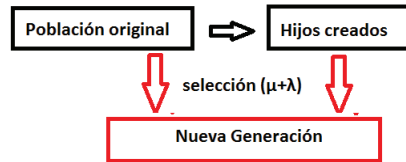
2.3.6.1. Selección fitness proporcional (μ, λ) :

En este caso, la siguiente generación es creada con el método de selección **fitness proporcional** (ver 2.2.5.1) tomando en cuenta sólo la población de hijos (y sus respectivos fitness).



2.3.6.2. Selección fitness proporcional ($\mu + \lambda$) :

En este caso, la siguiente generación es creada con el método de selección **fitness proporcional** (ver 2.2.5.1) tomando en cuenta la población de hijos y la generación actual.



2.3.6.3. Selección con elitismo

Por último, también se tiene que tomar en cuenta un **mecanismo adicional** de selección, el cual se denomina **elitismo**. Este mecanismo, como su nombre lo indica, selecciona automáticamente a los mejores k individuos de la generación actual que pasan a la siguiente generación (k número de individuos élitos). Los restantes $\mu - k$ individuos son seleccionados utilizando el método de fitness proporcional.

El **elitismo** es un mecanismo que nos permite tener la seguridad de que no se pierda el mejor individuo en la transición de las generaciones.

2.4. Estrategias evolutivas para la optimización de funciones

En esta sección se describe el algoritmo de estrategias evolutivas. Como en la sección pasada, nos concentramos en el problema de minimizar una función:

$$\phi : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}^+$$

Suponiendo que tenemos $D = [I_1, S_1] \times [I_2, S_2] \times \dots \times [I_n, S_n]$.

Como en la sección anterior, presentaremos en una primera instancia dos tipos de representaciones para atacar este problema. Después, presentaremos los distintos tipos de operadores vacacionales (mutación y recombinación) para cada una de estas representaciones. Por último, se presentan los mecanismos de selección y reemplazamiento que utilizaremos para manejar la población de soluciones.

2.4.1. Representación

Las estrategias evolutivas se utilizan principalmente para el tipo de problema en el cual trabajamos, es decir, la optimización de una función de \mathbb{R}^n a \mathbb{R} . La **representación**

tación usada es la flotante (o real), se tiene que:

$$\text{Genotipos} = (x_1, x_2, \dots, x_n) \in D$$

Sin embargo, en este caso los **individuos contienen parámetros estratégicos**, en particular, parámetros que se relacionan con el operador de mutación. Estos parámetros estratégicos σ **representan el paso de mutación**. Una forma general de ver a los individuos para este algoritmo es:

$$(x_1, x_2, \dots, x_n, \sigma_1, \sigma_2, \dots, \sigma_{n_\sigma})$$

Donde $(x_1, x_2, \dots, x_n) \in D$, n_σ es el número de parámetros estratégicos y $(\sigma_1, \sigma_2, \dots, \sigma_{n_\sigma})$ es el **vector de los parámetros estratégicos**.

2.4.2. Mutación

La mutación en las estrategias evolutivas están basadas en variables aleatorias con distribución normal de esperanza 0 y varianza σ . A continuación veremos distintos tipos de mutaciones utilizando los parámetros estratégicos como varianzas de variables aleatorias normales.

2.4.2.1. Mutación con sólo un parámetro estratégico

En este caso, los individuos sólo tienen un parámetro estratégico, es decir, son de la forma:

$$(x_1, x_2, \dots, x_n, \sigma)$$

Donde $(x_1, x_2, \dots, x_n) \in D$.

Esta mutación es muy similar a la mutación con una variable normal vista en los AG. Tomemos otra vez a:

$$H_i : \mathbb{R} \longrightarrow [I_i, S_i]$$

$$H_i(x) = \begin{cases} x, & \text{si } I_i \leq x \leq S_i \\ I_i & \text{si } x < I_i \\ S_i, & \text{si } x > S_i \end{cases}$$

nuestro operador Mut está dado como:

$$Mut(x_1, x_2, \dots, x_n, \sigma) = (H_1(x_1 + N_1), H_2(x_2 + N_2), \dots, H_n(x_n + N_n), \sigma * exp(N(0, \tau)))$$

Nota : $exp(N(0, \tau))$ es una variable log normal con parámetros $(0, \tau)$, y se le suele denominar $LogN(0, \tau)$.

Donde $\{N_i\}$ son variables aleatorias independientes con una función de distribución normal con parámetros $(0, \sigma)$.

Observación. En el contexto de las estrategias evolutivas, a τ se le conoce como el coeficiente de aprendizaje, en lo general se toma a τ bastante pequeño.

2.4.2.2. Mutación con n parámetros estratégicos

En este caso los individuos tienen n parámetros estratégicos, es decir, son de la forma:

$$(\underline{x}, \underline{\sigma})$$

Donde $\underline{x} = (x_1, x_2, \dots, x_n) \in D$ y $\underline{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_n) \in \mathbb{R}_+^n$.

Retomando la mutación anterior, tomemos:

$$Mut(\underline{x}, \underline{\sigma}) = (H(\underline{x} + \underline{N}), \underline{\sigma} * exp(\underline{N}))$$

Donde $H(\underline{x} + \underline{N}) = (H_1(x_1 + N_1), H_2(x_2 + N_2), \dots, H_n(x_n + N_n))$ y $\underline{\sigma} * exp(\underline{N}) = (\sigma_1 * exp(N(0, \tau)), \sigma_2 * exp(N(0, \tau)), \dots, \sigma_n * exp(N(0, \tau)))$.

Nótese que ahora las variables aleatorias normales $\{N_i\}$ se toman con parámetros $(0, \sigma_i)$.

La idea detrás de las mutaciones anteriormente presentadas es perturbar a los individuos y recordar su paso de perturbación para poder conservar su radio de movimiento. Heurísticamente se espera que las mutaciones sean bastante grandes al principio del algoritmo, debido a las variables aleatorias log normales, este período puede ser visto como el de exploración. Cuando los individuos se encuentren cerca de un mínimo (o máximo), se castigarán a los individuos que tienen un paso de perturbación grandes, puesto que esos pasos los alejarán del óptimo, y a su vez se premiarán a los que tienen pasos chicos, ya que estos se quedan cerca del punto deseado.

2.4.3. Recombinación

En las estrategias evolutivas, la recombinación toma dos padres para crear un sólo hijo a diferencia de los AG. Existen dos tipos de variantes para la recombinación: una llamada recombinación intermediaria y otra llamada recombinación discreta.

2.4.3.1. Recombinación discreta

La recombinación discreta se basa en seleccionar aleatoriamente coordenadas del padre y de la madre (equiprobablemente) para formar el hijo. Para describir este

Nota : $exp(N(0, \tau))$ es una variable log normal con parámetros $(0, \tau)$, y se le suele denominar $LogN(0, \tau)$.

procedimiento, tomemos el siguiente operador estocástico:

$$Rand : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$$

$$Rand(x, y) = \begin{cases} x & \text{si } U(0, 1) \leq \frac{1}{2} \\ y & \text{si } U(0, 1) > \frac{1}{2} \end{cases}$$

Donde $U(0, 1)$ representa una variable aleatoria uniforme en el intervalo $(0, 1)$.

El operador *Reco* está dado de la siguiente manera:

$$Reco(x, y) = (Rand(x_1, y_1), \dots, Rand(x_n, y_n), Rand(\sigma_{x1}, \sigma_{y1}), \dots, Rand(\sigma_{xn}, \sigma_{yn}))$$

2.4.3.2. Recombinación intermedia

La recombinación intermedia es simplemente una recombinación aritmética de los dos padres con un parámetro de proporción $\alpha = \frac{1}{2}$. Es decir, que nuestro operador *Reco* está dado como :

$$Reco : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

$$Reco(x, y) = \frac{x + y}{2}$$

(Nota: Estos métodos también suelen ser extendidos a más de dos padres, y suelen llamarse recombinaciones globales).

2.4.4. Mecanismos de selección de Padres

En las estrategias evolutivas la **selección de padres no se basa en el fitness**. Aquí, la selección es equiprobable para todos los individuos. En las estrategias evolutivas toda la **población es vista como una "población progenitora"**, a diferencia que en los AG donde el término padre designa a los individuos que fueron especialmente seleccionados para ser progenitores. Para concretar la idea podemos decir:

$$P\{x \text{ sea seleccionado para ser padre}\} = \frac{1}{\mu}$$

Donde $x \in Pob$ y $\mu = \#(Poblacion)$ es una constante que representa el número de individuos que hay en la población.

Nota : $exp(N(0, \tau))$ es una variable log normal con parámetros $(0, \tau)$, y se le suele denominar $LogN(0, \tau)$.

2.4.5. Mecanismos de selección de supervivientes

Recordemos que se denomina λ a la **constante que representa el número de hijos creados**, y μ a la **constante que representa el tamaño de la población**.

En los AE la generación siguiente suele remplazarse por los hijos o gran parte de ellos conforma la siguiente generación. Por esto mismo, se pide que $\lambda > \mu$.

2.4.5.1. Mecanismo de selección (μ, λ)

En este caso, seleccionan a los **mejores μ hijos para que conformen a la siguiente generación**. Es decir, que el operador Sel puede ser presentado como:

$$Sel : \mathbb{R}^{n \times \lambda} \longrightarrow \mathbb{R}^{n \times \mu}$$

$$Sel(x_1, x_2, \dots, x_\lambda) = (x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(\mu)})$$

Donde $\pi(1), \pi(2), \dots, \pi(\mu), \pi(\mu+1), \dots, \pi(\lambda)$ es una permutación de los índices $1, 2, \dots, \lambda$, tales que:

$$\phi(x_{\pi(1)}) \leq \phi(x_{\pi(2)}) \leq \dots \leq \phi(x_{\pi(\lambda)})$$

2.4.5.2. Mecanismo de selección $(\mu + \lambda)$

En este caso, seleccionan a los **mejores μ individuos tomando en cuenta la población μ de padres y población λ de hijos** para crear la siguiente generación. Es decir, que el operador Sel puede ser presentado como:

$$Sel : \mathbb{R}^{n \times (\mu + \lambda)} \longrightarrow \mathbb{R}^{n \times \mu}$$

$$Sel(x_1, x_2, \dots, x_{\mu + \lambda}) = (x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(\mu)})$$

Donde $\pi(1), \pi(2), \dots, \pi(\mu + \lambda)$ es una permutación de los índices $1, 2, \dots, \mu + \lambda$, tales que:

$$\phi(x_{\pi(1)}) \leq \phi(x_{\pi(2)}) \leq \dots \leq \phi(x_{\pi(\mu + \lambda)})$$

Cabe notar que con esta selección se garantiza que si el mejor individuo se encuentra en la generación de padres, entonces pasará a la siguiente generación.

Nota : $exp(N(0, \tau))$ es una variable log normal con parámetros $(0, \tau)$, y se le suele denominar $LogN(0, \tau)$.

3 Estimación de Parámetros en EDO's con aplicaciones

Resulta con frecuencia en la investigación y las aplicaciones que la función matemática está dada de manera implícita. Por ejemplo, como la solución de un problema de ecuaciones diferenciales ordinarias (EDOs) con valores iniciales, pero con parámetros desconocidos.

En este capítulo veremos una parte de la teoría de la estimación de parámetros en (EDOs), seguido por una breve presentación de un software desarrollado específicamente para el problema de estimación. Por último, veremos aplicaciones con casos reales y simulados.

Referencias del capítulo: [ALR],[ARN],[BRD],[KSB]

3.1. El problema de valores iniciales para EDOs

3.1.1. Formulación del problema (PVI) para EDOs

Se llama problema de valor inicial (PVI) o problema de Cauchy en un intervalo $[t_0, t_f]$, al dado por una EDO y una condición inicial en la forma:

$$\begin{cases} y' = f(t, y, \theta) \\ y(t_0) = \eta \end{cases} \quad (\text{condición inicial}) \quad (3.1)$$

Donde $t \in [t_0, t_f] \subset \mathbb{R}$ se le denomina el tiempo o variable de control, $y \in \mathbb{R}^m$ se le llama vector de estados del sistema, $\theta \in \mathbb{R}^p$ es vector de los parámetros.

3.1.2. Dos teoremas fundamentales para el estudio de parámetros de EDOs

A continuación presentaremos dos teoremas que nos permiten el estudio de los parámetros de las EDO's.

Observación. La prueba de estos teoremas es muy extensa, por esto mismo no se incluirán en esta tesis, sin embargo, el lector puede consultarlas en [?].

Teorema de existencia y unicidad[ARN]

Teorema 1. Si $f(t, y, \theta)$ es continua y globalmente Lipschitz con respecto a y sobre $[t_0, t_f] \times \mathbb{R}^m$ para θ fijo, entonces se tiene que existe una única solución de (3.1) en $[t_0, t_f]$ dada por:

$$F(t; t_0, \eta, \theta) = \eta + \int_{t_0}^t f(s, F(s; t_0, \eta, \theta), \theta) ds \quad t \in [t_0, t_f]$$

Este teorema nos indica los supuestos necesario para que la solución de (3.1) sea única y exista, lo cual es esencial para hacer estimación.

Teorema de dependencia continua con respecto a los parámetros y condiciones iniciales[ARN]

Teorema 2. Si además de las hipótesis del teorema 1 se tiene que $f(t, y, \theta)$ es Lipschitz con respecto a θ en $[t_0, T] \times G \times B_r(\theta_0)$, con $G \subset \mathbb{R}^m$, ($r > 0$) entonces la solución $F(t, t_0, \eta, \theta)$ de (3.1) es continua respecto a η y θ en $[t_0, T] \times G \times B_r(\theta_0)$.

Este teorema nos da las condiciones necesarias para que nuestra solución sea continua con respecto a los parámetros y las condiciones iniciales, lo cual es de vital importancia cuando se optimiza una función.

3.2. El problema de estimación de parámetros en EDOs

Consideramos un modelo dado como un (PVI) el cual tiene la forma de (3.1):

$$\begin{cases} y' = f(t, y, \theta) \\ y(t_0) = \eta \end{cases} \quad (\text{condición inicial})$$

Donde $t \in [t_0, t_f] \subset \mathbb{R}$, $y \in \mathbb{R}^m$, $\theta \in \mathbb{R}^p$.

Y unos datos de la forma:

$$D := \{(t_i, y_i)\}_{i \in \{0, 2, \dots, n\}} \subset \mathbb{R} \times \mathbb{R}^k$$

Donde $t_i \in [t_0, t_f] \subset \mathbb{R}$ y $y_i \in \mathbb{R}^k$ con $k \leq m$.

Supondremos que existen los parámetros reales del fenómeno estudiado, a este vector lo entenderemos por $\theta^* \in \mathbb{R}^p$, o como $(\theta^*, \eta^*) \in \mathbb{R}^p \times \mathbb{R}^m$ si las condiciones iniciales son vistas como parámetros. Por lo tanto, se tiene que si $F(t, \eta, \theta)$ es solución de (3.1) entonces:

$$F_j(t, \eta^*, \theta^*) = y_{ij} + \epsilon_{ij} \quad j = 1, 2, \dots, k \quad (3.2)$$

Donde $F_j(t, \eta, \theta)$, y_{ij} son las proyecciones de la j -ésima coordenada de $F(t, \eta, \theta)$, y_i , respectivamente, y donde ϵ_{ij} el error de medición del i_j -ésimo dato.

Nos interesa precisamente encontrar a θ^* ó (θ^*, η^*) , para lo cual abordaremos este problema como uno de optimización.

3.2.1. Métodos de estimación de parámetros en EDOs

Sea $f(t, y, \theta)$ de manera que satisfaga las condiciones del teorema 1 y sea $F(t, \eta, \theta)$ solución de (3.1). En lo general, se denotará a $F(t, \eta, \theta)$ como $F(t, \theta)$.

3.2.1.1. Diferenciación de los datos

Una aproximación de las derivadas que aparecen en la ecuación, podría ser adquirida diferenciando los datos adyacentes. Suponiendo $k = m$ (es decir, que hay datos de todas las variables de estado), se puede decir que:

$$\frac{(y_{i+1}) - (y_{i-1})}{t_{i+1} - t_{i-1}} \simeq f(t_i, y_i, \theta^*) \quad (3.3)$$

Por lo que tomaríamos el siguiente funcional:

$$\phi(\theta) = \sum_{i=1}^n \left\| \left(\frac{(y_{i+1}) - (y_{i-1})}{t_{i+1} - t_{i-1}} \right) - f(t_i, y_i, \theta) \right\|_2^2$$

Observación. Nótese que si f es lineal sobre θ , entonces ¡Esto se reduce al caso lineal del capítulo 1 (1.33)!

Para apreciar (3.3) veamos que de (3.2) se tiene que:

$$\begin{aligned} \frac{(y_{i+1}) - (y_{i-1})}{t_{i+1} - t_{i-1}} &= \frac{[F(t_{i+1}, \theta^*) - F(t_{i-1}, \theta^*)] - [\epsilon_{i+1} - \epsilon_{i-1}]}{t_{i+1} - t_{i-1}} \\ &= \frac{[F(t_{i+1}, \theta^*) - F(t_{i-1}, \theta^*)]}{t_{i+1} - t_{i-1}} + \frac{[\epsilon_{i+1} - \epsilon_{i-1}]}{t_{i+1} - t_{i-1}} \end{aligned} \quad (3.4)$$

Dada $\delta_2 > 0$, se puede probar fácilmente, bajo el supuesto de que $F \in C^2$, que existe $\delta_1 > 0$ tal que si $|t_{i+1} - t_{i-1}| \leq \delta_1$ entonces:

$$\left\| \frac{[F(t_{i+1}, \theta^*) - F(t_{i-1}, \theta^*)]}{t_{i+1} - t_{i-1}} - f(t_i, y_i, \theta^*) \right\|_2^2 \leq \delta_2 \quad (3.5)$$

Definimos a δ_3 como:

$$\left\| \frac{[\epsilon_{i+1} - \epsilon_{i-1}]}{\delta_1} \right\|_2^2 = \delta_3 \quad (3.6)$$

Juntando las (3.4) , (3.5) y (3.6):

$$\left\| \frac{(y_{i+1}) - (y_{i-1})}{t_{i+1} - t_{i-1}} - f(t_i, y_i, \theta^*) \right\|_2^2 \leq \delta_2 + \delta_3 \quad (3.7)$$

Tomando a δ_2 bastante pequeño y suponiendo que las **mediciones son extremadamente precisas o que** $\|\epsilon_{i+1} - \epsilon_{i-1}\| \leq |t_{i+1} - t_{i-1}|^2$ se puede decir que δ_3 es pequeño, por lo que:

$$\frac{(y_{i+1}) - (y_{i-1})}{t_{i+1} - t_{i-1}} \simeq f(t_i, y_i, \theta^*)$$

Esto nos implica, en particular, que:

$$\begin{aligned} \phi(\theta) &= \sum_{i=1}^n \left\| \left(\frac{(y_{i+1}) - (y_{i-1})}{t_{i+1} - t_{i-1}} \right) - f(t_i, y_i, \theta) \right\|_2^2 \leq n(\delta_2 + \delta_3) \\ &\simeq \sum_{i=1}^n \|f(t_i, y_i, \theta^*) - f(t_i, y_i, \theta)\|_2^2 \end{aligned}$$

(Donde la última parte se garantiza solamente si $n(\delta_2 + \delta_3)$ es bastante pequeño).

Observación. Si no se conoce la condición inicial se toma a ϕ como:

$$\phi(\theta, \eta) = \left\| \left(\frac{(y_2) - (\eta)}{t_{i+1} - t_{i-1}} \right) - f(t_i, y_i, \theta) \right\|_2^2 + \sum_{i=2}^n \left\| \left(\frac{(y_{i+1}) - (y_{i-1})}{t_{i+1} - t_{i-1}} \right) - f(t_i, y_i, \theta) \right\|_2^2$$

La ventaja principal de este método es que se puede utilizar directamente la ecuación (3.1) para calcular el error, y por lo tanto el cómputo de la función objetivo es mucho más rápido y sencillo.

Sin embargo, este método tiene sus inconvenientes, puesto que se necesita un conjunto de datos **denso y extremadamente preciso (eq 3.6)**, más aún se tiene que tener datos de todas las variables de estado, en la práctica esto no pasa frecuentemente, a menos que sea un experimento especialmente planeado.

3.2.1.2. Integración de los datos

Se aproxima la función solución $F(t, \theta^*)$ integrando los datos. Con este fin, integramos de ambos lados la ecuación (3.1), la ecuación diferencial se transforma en una ecuación integral, dada por:

$$\begin{aligned} \int_{t_0}^{t_i} f(t, y, \theta^*) dt &= F(t_i, \theta^*) - F(t_0, \theta^*) + [\epsilon_i - \epsilon_0] - [\epsilon_i - \epsilon_0] \\ &= y_i - y_0 - [\epsilon_i - \epsilon_0] \end{aligned}$$

Ya que $y_i = F(t_i, \theta^*) + \epsilon_i$ (ϵ_i es el vector de error de mediciones).

Por otro lado, utilizando regla del trapecoide:

$$\begin{aligned} \int_{t_0}^{t_i} f(t, y, \theta^*) &= \frac{1}{2} \left(\sum_{j=1}^i (f(t_j, F(t_j, \theta^*), \theta^*) + f(t_{j-1}, F(t_{j-1}, \theta^*), \theta^*)) (t_j - t_{j-1}) \right) \\ &= \frac{1}{2} \left(\sum_{j=1}^i (f(t_j, y_j - \epsilon_j, \theta^*) + f(t_{j-1}, y_{j-1} - \epsilon_{j-1}, \theta^*)) (t_j - t_{j-1}) \right) \end{aligned} \quad (3.8)$$

Como f es continua en la variable de control y , es posible decir que $f(t_j, y_i - \epsilon_i, \theta^*) = f(t_j, y_j, \theta^*) + \delta_i$ siempre y cuando ϵ_i sea bastante pequeño. Por lo que (3.8) se transforma en :

$$\begin{aligned} \int_{t_1}^{t_i} f(t, y, \theta^*) &= \frac{1}{2} \left(\sum_{j=1}^i (f(t_j, y_j, \theta^*) + f(t_{j-1}, y_{j-1}, \theta^*)) (t_j - t_{j-1}) \right) + \frac{1}{2} \left(\sum_{j=1}^i (\delta_i + \delta_{j-1}) (t_j - t_{j-1}) \right) \\ &= \frac{1}{2} \left(\sum_{j=1}^i (f(t_j, y_j, \theta^*) + f(t_{j-1}, y_{j-1}, \theta^*)) (t_j - t_{j-1}) \right) + \delta \end{aligned} \quad (3.9)$$

Donde $\delta = \frac{1}{2} \left(\sum_{j=1}^i (\delta_i + \delta_{j-1}) (t_j - t_{j-1}) \right)$, si $|t_j - t_{j-1}|$ es bastante pequeño a comparación de $|\delta_i + \delta_{j-1}|$ entonces, δ es bastante pequeño.

Juntando (3.8) y (3.9) se tiene:

$$y_i - y_0 = \frac{1}{2} \left(\sum_{j=1}^i (f(t_j, y_j, \theta^*) + f(t_{j-1}, y_{j-1}, \theta^*)) (t_j - t_{j-1}) \right) + \delta - [\epsilon_i - \epsilon_0]$$

Por lo que si $[\epsilon_i - \epsilon_0]$ es bastante pequeño se deduce que:

$$y_i - y_0 \simeq \frac{1}{2} \left(\sum_{j=1}^i (f(t_j, y_j, \theta^*) + f(t_{j-1}, y_{j-1}, \theta^*)) (t_j - t_{j-1}) \right)$$

Definiendo a $I_i(\theta) = \frac{1}{2} \left(\sum_{j=1}^i (f(t_j, y_j, \theta) + f(t_{j-1}, y_{j-1}, \theta)) (t_j - t_{j-1}) \right)$, claramente se tiene que $I_i(\theta^*) \simeq \int_{t_1}^{t_i} f(t, y, \theta^*) dt$. Por lo tanto, nuestra función objetivo sería:

$$\begin{aligned}\phi(\theta) &= \sum_{i=1}^n [y_i - y_1 - I_i(\theta)]^2 \\ &\simeq \sum_{i=1}^n [I_i(\theta^*) - I_i(\theta)]^2\end{aligned}$$

Observación. Si no se conoce la condición inicial se toma a ϕ como:

$$\phi(\theta, \eta) = \sum_{i=2}^{M+1} \|y_i - \eta - I_i(\theta)\|_2^2$$

Hay que notar que este método es más estable que el de diferenciación de datos, puesto que en lo general la integración de los datos resulta ser menos susceptible a perturbaciones que la diferenciación (δ eq.(3.9)), pero aún así es difícil poder controlar el error obtenido de este método, además, también se necesita un conjunto bastante denso de datos. Sin embargo, una ventaja sería que el cómputo de este método es bastante rápido.

3.2.1.3. Integración de la ecuación diferencial

Sabemos que la solución de (3.1) tiene la forma :

$$y = F(t, \eta, \theta)$$

Lo cual nos lleva a tratar el problema de optimización exactamente igual que en el capítulo 1:

$$\phi(\theta, \eta) = \sum_{i=0}^n [y_i - F(t_i, \eta, \theta)]^2$$

Observación. Para este caso se tiene que tomar a f satisfaciendo como mínimo las condiciones del teorema 2, para que de esta manera se pueda garantizar que ϕ es continua.

Si la ecuación (3.1) tiene una solución analítica podemos olvidarnos de la ecuación diferencial y optimizar simplemente la función solución. Generalmente no se puede encontrar este tipo de solución, y para resolver este problema se está obligado a integrar numéricamente la ecuación diferencial utilizando métodos conocidos (por ejemplo, Runge-Kutta).

La desventaja de este método son las condiciones iniciales que se necesitan para resolver la ecuación, cuando estas no se conocen se deben poner como parte de los parámetros que se desean encontrar, también el tiempo de cómputo es mayor

puesto que en cada iteración de un algoritmo se tiene que calcular numéricamente las soluciones.

Una ventaja, sin embargo, es que no se necesitan tantos datos como en los métodos anteriores, y además en la mayoría de los métodos de integración se puede controlar el error absoluto y relativo de este procedimiento, esto mismo aumenta la precisión de los parámetros estimados.

3.2.2. Ejemplo de aplicación de los métodos

Ilustraremos a continuación los distintos métodos con un ejemplo bastante sencillo y académico. Supongamos que tenemos el siguiente modelo:

$$\begin{cases} \frac{dy}{dt} = \theta_1 y \\ y(t_0) = \eta \end{cases} \quad (3.10)$$

Donde $y \in \mathbb{R}$, $\theta \in \mathbb{R}$; y el siguiente conjunto de datos:

$$D := \{(t_i, y_i)\}_{i \in \{0, 2, \dots, n\}}$$

3.2.2.1. Diferenciación de los datos:

Conociendo a las condiciones inicial, es decir, tomando a $c = y_1$ en la ecuación (1.2), definamos:

$$u_i = \frac{(y_{i+1}) - (y_{i-1}))}{t_{i+1} - t_{i-1}}$$

Se puede ver que u_i es una aproximación de $y'(t_i)$. Por lo tanto, la estimación se basaría, por ejemplo, en la optimización de la función de mínimos cuadrados:

$$\phi(\theta) = \sum_{i=1}^{n-1} [u_i - \theta y_i]^2$$

Si no se conocen las condiciones iniciales entonces se tiene:

$$\phi(\theta, \eta) = \left[\frac{(y_2) - (\eta)}{t_2 - t_0} - \theta y_1 \right]^2 + \sum_{i=2}^{n-1} [u_i - \theta y_i]^2$$

3.2.2.2. Integración de los datos:

Para aplicar este método primero transformemos la ecuación diferencial (3.1) a una ecuación integral, es decir, integremos de ambos lados de t_0 a t_i , y por lo tanto

obtenemos:

$$y(t_i) - y(t_0) - \theta \int_{t_0}^{t_i} y(t) dt = 0$$

Definamos:

$$I_i(\theta) = \frac{1}{2} \left(\theta \sum_{j=1}^i (y_j + y_{j-1})(t_j - t_{j-1}) \right)$$

Y entonces, se sigue que la función de mínimos cuadrados asociada al problema de estimación sería:

$$\begin{aligned} \phi(\theta) &= \sum_{i=1}^n [y_i - y_0 - I_i(\theta)]^2 \\ &= \sum_{i=1}^n [y_i - y_0 - \theta I_i]^2 \end{aligned}$$

Suponiendo que no conocemos la condición inicial, tenemos:

$$\phi(\theta, \eta) = \sum_{i=2}^{M+1} [y_i - \eta - I_i(\theta)]^2$$

3.2.2.3. Integración de la ecuación diferencial:

Integrando obtenemos:

$$y = y(t_0) \exp(\theta t)$$

Suponiendo que no conocemos la condición inicial, es decir $y(t_0)$, la estimación de los parámetros se haría optimizando la siguiente función:

$$\phi(\theta, \eta) = \sum_{i=0}^n [y_i - \eta \exp(\theta t)]^2$$

Para el programa desarrollado en este trabajo se decidió utilizar la integración de la ecuación diferencial, dado que nuestro propósito es brindar una herramienta en un cierto sentido genérico, tenemos que presuponer las peores condiciones posibles sobre los datos que se trabajan, además hoy en día la capacidad de cómputo ha incrementado significativamente. Por lo tanto, aunque sí es importante, tampoco es fundamental preocuparse demasiado en la capacidad de cómputo, a menos que sean problemas muy específicos con una cantidad de datos y ecuaciones exorbitantes.

3.3. Programa de estimación de parámetros para EDO's

3.3.1. Algoritmos y métodos utilizados

Para el desarrollo del programa se utilizó el **método de la integración de la ecuación diferencial** visto en la subsección 3.1.3, con una pequeña modificación. En lo general, se minimiza el siguiente funcional:

$$\phi(\theta, \eta) = \sum_{i=0}^n \frac{[y_i - F(t_i, \eta, \theta)]^2}{|y_i| + \epsilon}$$

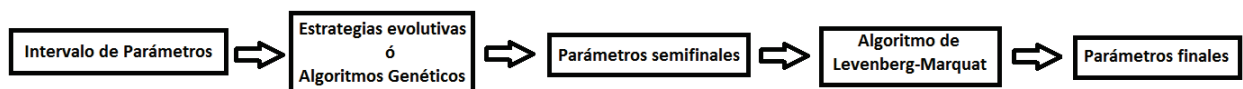
Donde $\{(t_i, y_i)\}_{i \in \{0,1,2,\dots,n\}} \subset \mathbb{R} \times \mathbb{R}^k$ es nuestro conjunto de datos y $y = F(t, \eta, \theta)$ es la solución de la ecuación (3.1).

Para el proceso de minimización primero utilizamos los algoritmos evolutivos con los siguientes esquemas (depende de la selección de los usuarios):

Estrategias Evolutivas:	
Representación	vectores reales
Mutación	n parámetros estratégicos
Recombinación	Discreta o intermedia
Selección de Padres	uniforme
selección de supervivientes	$(\mu + \lambda)$

Algoritmos Genéticos:	
Representación	vector bits
Mutación	bit-flip
Recombinación	1-crossover
Selección de Padres	fitness proporcional
selección de supervivientes	$(\mu + \lambda)$

Después de obtener una aproximación del mínimo con alguno de estos algoritmos se utiliza el algoritmo de Levenberg-Marquat, visto en el primer capítulo.



3.3.2. Detalles técnicos

3.3.2.1. Lenguaje de programación

El programa fue hecho utilizando el lenguaje de programación C++, para la interfaz del programa se utilizó el programa Qt.5.02.

3.3.2.2. Bibliotecas usadas

Para leer los modelos se utilizó la biblioteca libre *muparser* creada por alumnos del MIT. El sitio web de ésta es:

<http://muparser.beltoforion.de/>

Para resolver las ecuaciones diferenciales se utilizó la biblioteca libre *odeint* que contiene varios tipos de integradores:

<http://headmyshoulder.github.io/odeint-v2/>

Los algoritmos evolutivos usados se programaron manualmente con base al libro [ESM].

El algoritmo de Leven-bermarquart se basó en el código libre *lmfit* una versión de MINPACK, la cual puede obtenerse en:

<http://apps.jcns.fz-juelich.de/doku/sc/lmfit>

3.3.3. Sitio del programa

El siguiente enlace contiene los tutoriales del programa y su respectivo link de descarga. Actualmente sólo está disponible para Windows pero se puede compilar también en Linux.

<https://sites.google.com/a/ciencias.unam.mx/estimador-de-parametros/>

3.3.3.1. Links de los tutoriales del programa

Sitio de los tutoriales: <https://sites.google.com/a/ciencias.unam.mx/estimador-de-parametros/manual-de-usuario>

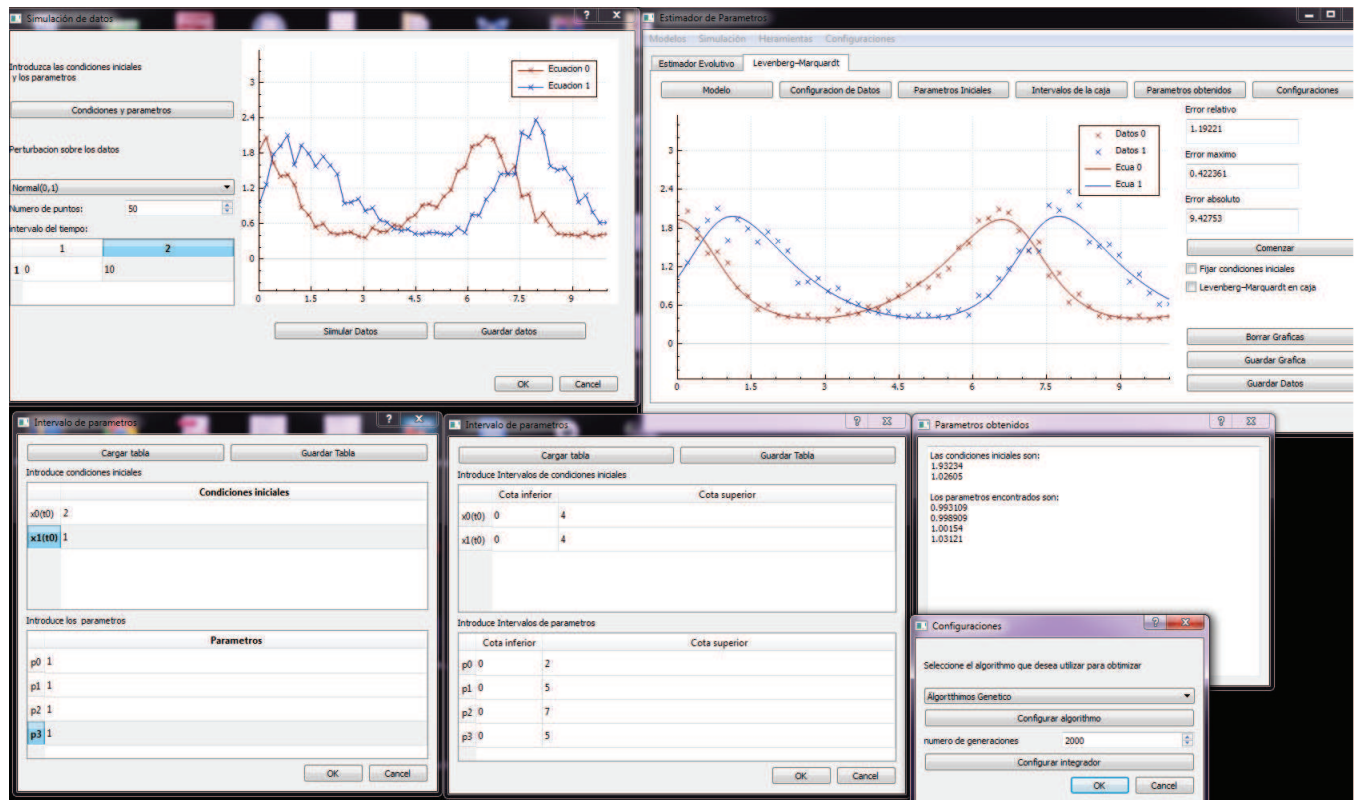
Sitio de los tutoriales en video: <https://www.youtube.com/playlist?list=PLMvY5X0139gW9X4aM1K-2yj-mhm>

3.3.3.2. Links de descarga del programa

La versión actual (3.0) sólo está disponible para Windows pero se puede compilar también en Linux, esto será posible cuando se libere el código fuente, lo cual sucederá después de que se presente esta tesis.

Link de descarga del programa: <https://sites.google.com/a/ciencias.unam.mx/estimador-de-parametros/documents>

3.3.4. Vista previa del programa



3.4. Aplicaciones de la estimación de parámetros de EDO's.

3.4.1. Estimación de parámetros artificiales

Para validar nuestro programa vamos a construir sistemas de prueba con datos artificiales. Se tomarán ecuaciones diferenciales de la forma:

$$\begin{cases} y' = f(t, y, \theta) \\ y(t_0) = \eta \end{cases} \quad (\text{condición inicial}) \quad (3.11)$$

Donde $t \in [t_0, t_f] \subset \mathbb{R}$ es la variable de control, $y \in \mathbb{R}^m$ el vector de estados del sistema, $\theta \in \mathbb{R}^p$ el vector de los parámetros.

Las cuales resolveremos con la técnica de Runge-Kutta de cuarto orden y posteriormente se perturbarán los datos utilizando el siguiente esquema:

$$y_p = y_e(1 + N(0, 1)) \quad (3.12)$$

Donde y_e son los datos obtenidos con el método de Runge-Kutta, $N(0, 1)$ es una variable normal con esperanza 0 y varianza 1, y_p son los datos perturbados que se usarán para la estimación con el programa anterior.

3.4.1.1. Sistema Lotka-Volterra

El sistema de Lotka-Volterra viene dado por las ecuaciones:

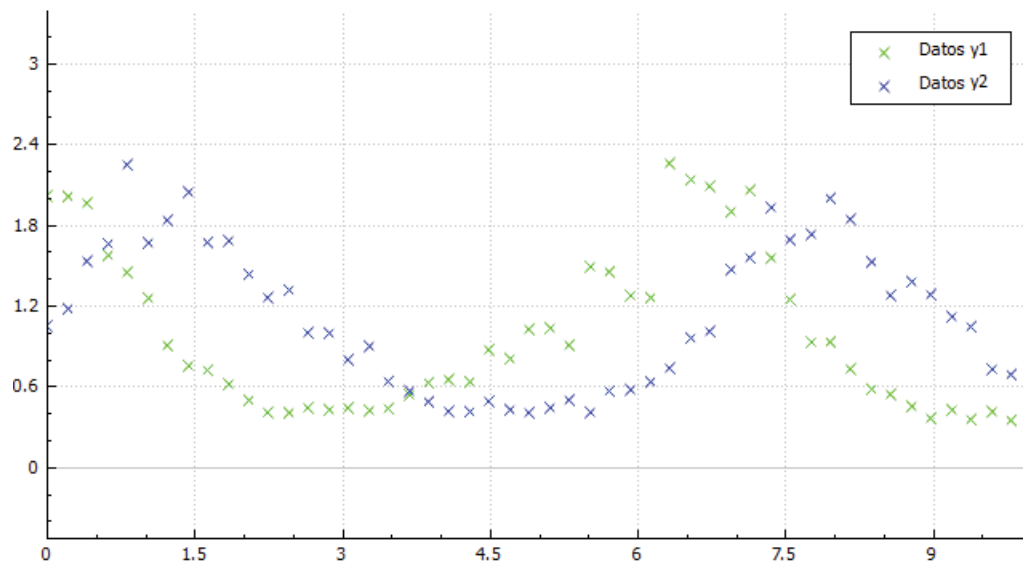
$$\begin{cases} \dot{y}_1 = \beta_1 y_0 - \beta_2 y_0 y_1 \\ \dot{y}_2 = \beta_3 y_0 y_1 - \beta_4 y_1 \end{cases}$$

Este sistema describe una interacción tipo presa-depredador, suponiendo que la razón de encuentros entre los individuos es proporcional al producto de las dos poblaciones.

Para nuestro experimento se generaron 100 observaciones (50 para cada variable de estado) en el intervalo $[0, 10]$. Se perturbaron los datos utilizando la ecuación (3.12).

Gráfica de los datos perturbados:

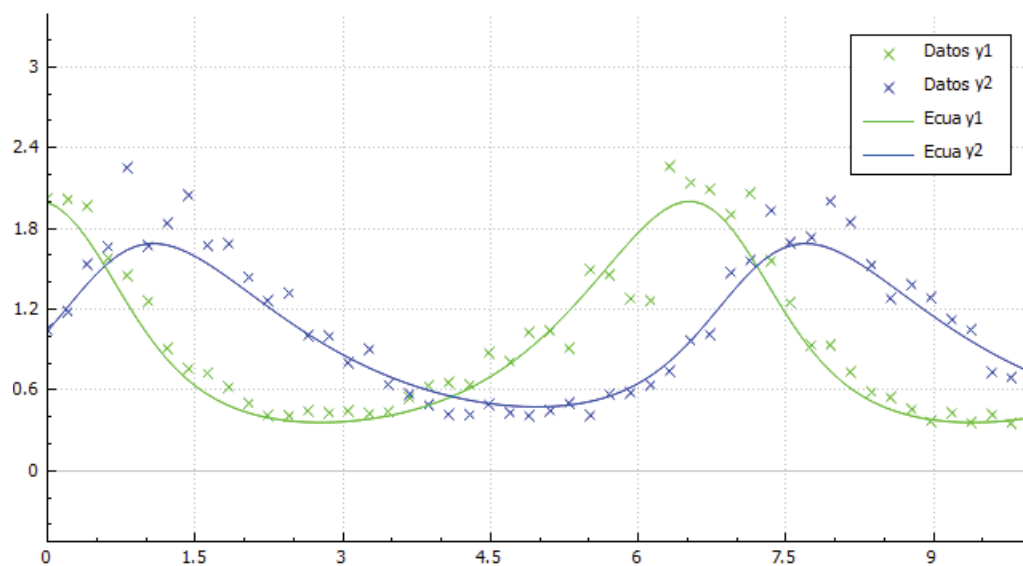
3.4 Aplicaciones de la estimación de parámetros de EDO's.



Los parámetros se buscan inicialmente en la caja:

Condición inicial	Cota inferior	Cota superior
$y_1(0)$	0	10
$y_2(0)$	0	10
Parámetro	Cota inferior	Cota superior
β_1	0	10
β_2	0	10
β_3	0	10
β_4	0	10

Gráfica de la estimación :



Resultados

Parámetros	$y_1(0)$	$y_2(0)$	β_1	β_2	β_3	β_4
Exactos	2.0	1.0	1.0	1.0	1.0	1.0
Estimados	1.9876	0.9656	1.0224	0.9714	1.0496	0.9752
Error %	0.62 %	3.44 %	2.24 %	2.86 %	4.96 %	2.48 %

Residual	1.2398
Error máximo	0.5267

3.4.1.2. Modelo SIR

Introducción En 1927, W. O. Kermac y A. G. Mckendrick publicaron un modelo matemático para el estudio del comportamiento de una plaga que azotó Bombay en los años 1905-1906, bajo los supuestos de modelación siguientes:¹

1. Una fracción de la población es contagiosa.
2. El individuo infectado adquiere inmunidad o muere.
3. El contagio de la infección obedece la ley de acción de masas.
4. Todas las personas susceptibles de la población tienen igual susceptibilidad para contraer la infección.
5. La población es cerrada, i.e., se desprecian movimientos migratorios.
6. El tamaño de la población es suficientemente grande.
7. La población es considerada constante, despreciando la dinámica demográfica, al tomar en cuenta que la epidemia dura unos pocos meses, como gripe e influenza, entre otras.

La población se divide en tres compartimentos: el de los individuos susceptibles de contraer la infección (S), el de los infectados y con capacidad de transmitir la enfermedad (I), y el de los recuperados (R) que son aquellos que adquieren inmunidad o mueren. “*La tasa de infección que determina el número de individuos por unidad de tiempo que se transfiere del compartimiento de susceptibles a infecciosos depende del número de contactos per cápita c que una persona sana susceptible tenga por unidad de tiempo y de la proporción ϕ de estos contactos que sean con individuos infecciosos que le transmitan la enfermedad*”.² Donde la tasa per cápita se puede denotar de la siguiente manera $\beta = \phi c$, dado el número de contactos realizados que de estos resulten efectivos, suponiendo que la tasa de infección es igual a βSI (i.e.,

¹La población permanece constante el tiempo t , no hay nacimientos, ni migración y fallecimientos, al menos, al momento de realizar el estudio.

²Jorge X. Velasco Hernandez, 2007

el número de individuos que pasa de la clase de los susceptibles a la clase de los infectados, por unidad de tiempo), y μ es la tasa per cápita de los recuperados o muertos. Luego, la tasa de los individuos que pasan de la clase de infectados a la de los recuperados es μI . El modelo, bajo los supuestos anteriores, es el siguiente:

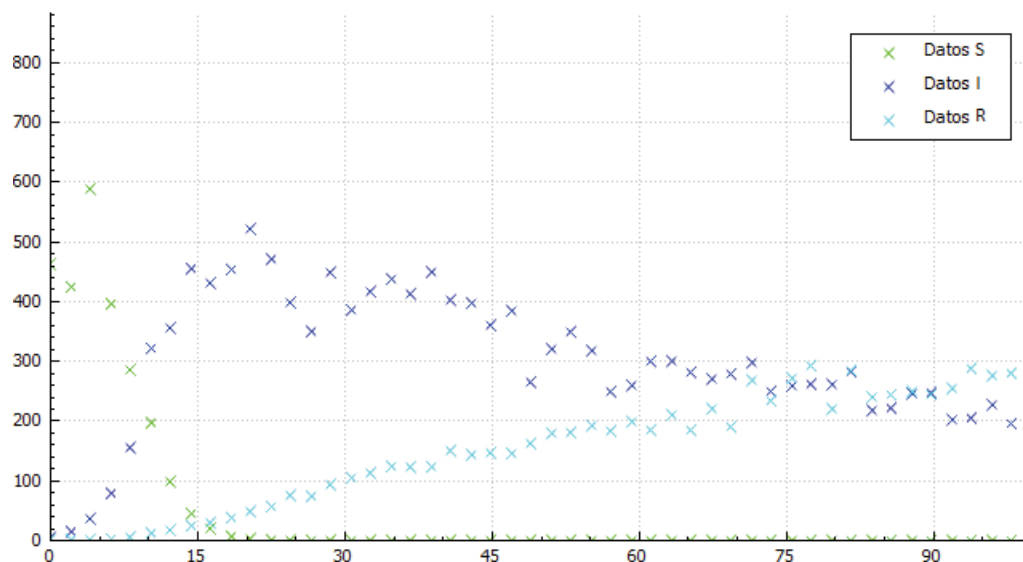
$$\begin{aligned} \dot{S} &= -\beta SI \\ \dot{I} &= \beta SI - \gamma I \\ \dot{R} &= \gamma I \end{aligned} \tag{3.13}$$

Otra manera de interpretar el modelo es considerar a β como la probabilidad de que un susceptible sea infectado y $1/\gamma$ es el tiempo promedio de la infección en un individuo.

Claramente, el dominio de definición de este sistema, con sentido biológico, es el octante positivo en \mathbb{R}^3 . Y para condiciones iniciales $S(t_0) = S_0 \geq 0$, $I(t_0) = I_0 \geq 0$ y $R(t_0) = R_0 \geq 0$ en $t_0 = 0$.

Estimación de los parámetros: Para nuestro experimento se generaron 100 observaciones (50 para los infectados y 50 para los recuperados) en el intervalo $[0,100]$. Se perturbaron los datos utilizando la ecuación (3.4).

Gráfica de los datos perturbados:

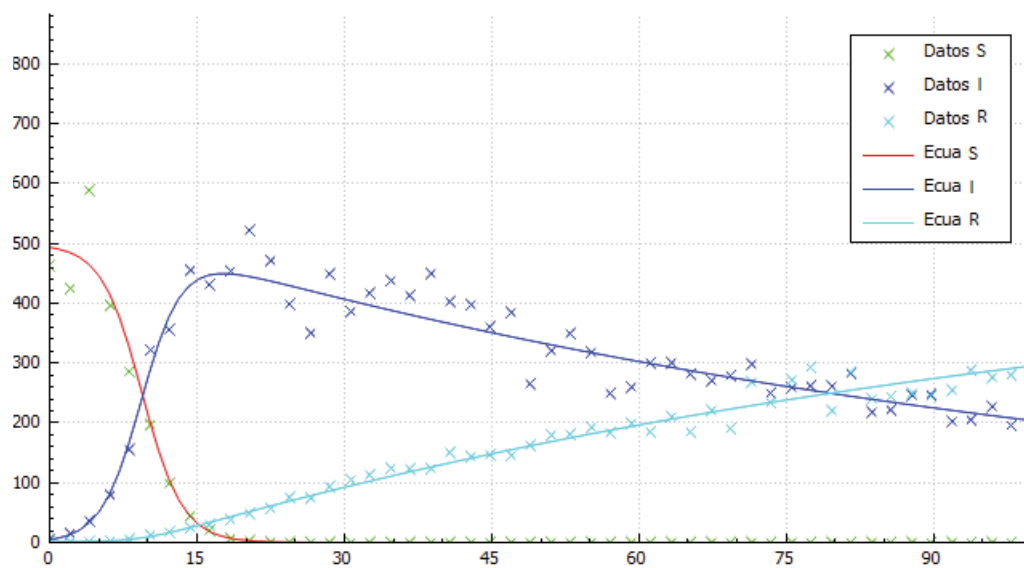


Observación. Los datos correspondientes a los susceptibles (“Datos S”) no se usaron para la estimación.

Los parámetros se buscan inicialmente en la caja:

Condición inicial	Cota inferior	Cota superior
$S(0)$	0	500
$I(0)$	0	15
$R(0)$	0	15
Parámetro	Cota inferior	Cota superior
β	0	1
γ	0	1

Gráfica de la estimación :



Resultados:

Parámetros	$S(0)$	$I(0)$	$R(0)$	β_2	γ
Exactos	500	5	1	0.001	0.01
Estimados	492.818	4.84781	1.03844	0.00100269	0.00987352
Error %	1.44 %	3.44 %	3.04 %	0.27 %	1.26 %

Residual	204.471
Error máximo	77.4185

3.4.2. Estimación de parámetros con datos reales

3.4.2.1. Población de México, modelo con la logística

En este caso, modelaremos la población de México utilizando la ecuación logística:

$$\frac{dP}{dt} = \beta P \left(1 - \frac{P}{K}\right) \quad (3.14)$$

Este modelo fue ideado por Verhulst y presupone que la tasa de cambio de la población es proporcional a una constante por la población, por 1 menos, la población entre la población de saturación.

Las variables que representan al modelo de la Ecuación 3.14 son:

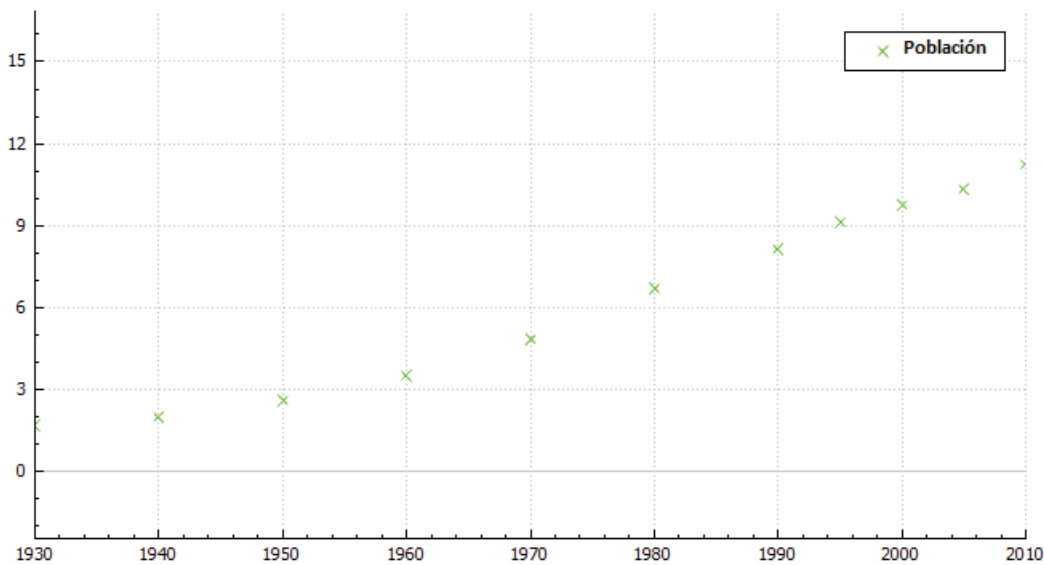
1. P representa al número de individuos al tiempo t .
2. K es el tamaño de la población de saturación.
3. β es la tasa de crecimiento per cápita.

La tabla de datos obtenida gracias al INEGI es la siguiente:

Años	1930	1940	1950	1960	1970	1980
Población en decenas de millones	1.65	1.96	2.57	3.49	4.82	6.68

Años	1990	1995	2000	2005	2010
Población en decenas de millones	8.12	9.11	9.74	10.32	11.23

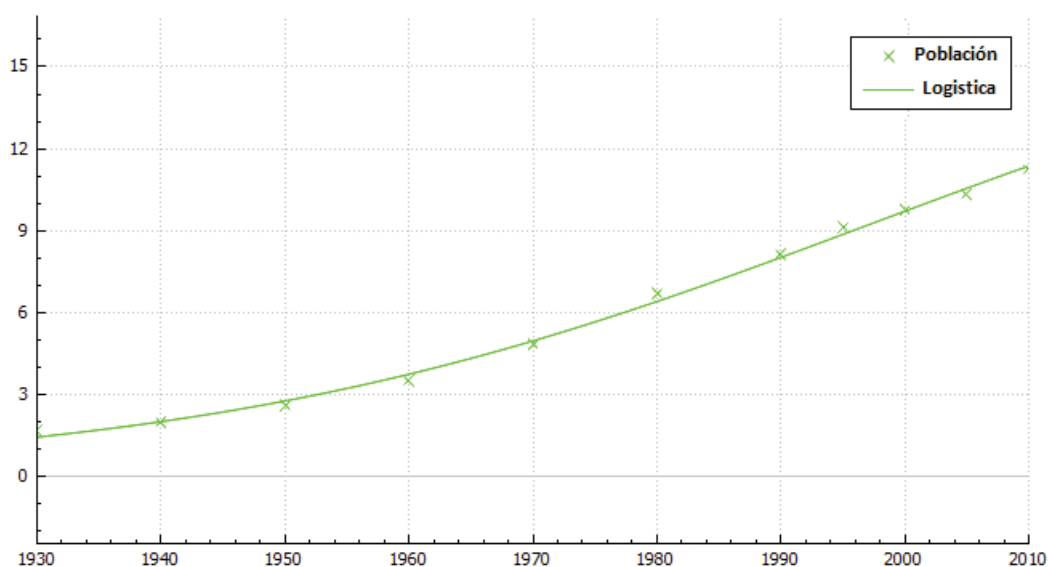
Gráfica de los datos:



Los parámetros se buscaron en la caja:

Condición inicial	Cota inferior	Cota superior
P_0	1.4	2
Parámetro	Cota inferior	Cota superior
β	0	1
K	10	20

Gráfica de la estimación :



Resultados:

Parámetros	P_0	β	K
Estimados	1.42	0.037	18.38

Residual	0.091542
Error máximo	0.295872

Como una pequeña conclusión se espera que el número máximo de habitantes en México sea del orden de 180 millones.

3.4.2.2. El caso de la obesidad en México con SIR modificado

Introducción Para el caso de epidemias cuyo periodo de desarrollo es de varias décadas, como es el caso del VIH/SIDA, Hepatitis C, Obesidad y Diabetes, no se pueden abstraer los efectos demográficos de la población. Este modelo a continuación describe el comportamiento de una población considerando efectos demográficos y abstrayendo la recuperación (curación) de los individuos infectados, bajo la hipótesis de que la población bajo estudio exhibe una población de saturación debido a la inhibición de la tasa per capita de crecimiento, i.e proporcional a $(1/S)$. Por lo anterior, el modelo está dado por:

$$\begin{aligned}\dot{S} &= \beta \left(1 - \frac{S}{K}\right) - \gamma SI \\ \dot{I} &= (\gamma S - \mu)I\end{aligned}\tag{3.15}$$

Donde:

1. $K = \beta/\nu$ es el tamaño de la población de saturación.
2. β es la tasa de crecimiento al tiempo t .
3. γ es la tasa de infección al tiempo t .
4. ν es la tasa per cápita de muerte de los susceptibles al tiempo t .
5. μ es la tasa de muerte per cápita de los infecciosos al tiempo t .

Estimación de los parámetros: Nuestro objetivo es estimar los parámetros para nuestro modelo (S,I). Sin embargo, para efectos de practicidad decidimos hacer un reescalamiento:

$$\begin{aligned}\sigma &= 10^{-8}S \\ \eta &= 10^{-7}I\end{aligned}$$

Esto se debe a que los susceptibles están dados en cientos de millones y los infectados en decenas de millones, de esta forma nuestro sistema es del orden de la unidad. Es fácil comprobar por el lector que nuestro nuevo modelo reescalado es el siguiente:

$$\begin{aligned}\dot{\sigma} &= \beta_s \left(1 - \frac{\sigma}{K_s}\right) - \frac{\gamma_s \sigma \eta}{10} \\ \dot{\eta} &= (\gamma_s \sigma - \mu)\eta\end{aligned}$$

Donde:

$$\begin{aligned}\beta_s &= 10^{-8}\beta \\ K_s &= 10^{-8}K \\ \gamma_s &= 10^8\gamma\end{aligned}$$

Recordemos que R_0 esta dado por:

$$R_0 = \frac{\gamma K}{\mu}$$

Entonces el número básico reproductivo de nuestro modelo reescalado es:

$$R_{0_s} = R_0$$

Gracias a la estimación anterior podemos dar una cota de β_s :

$$\beta_s \leq 0.02778$$

Como se puede ver en la tabla de la población en México, las tasas de crecimiento decrecen desde del año 2000. Es por esto mismo que tomamos como suposición que:

$$K_s \leq 18$$

Para la búsqueda de parámetros se utilizo en programa, en un principio los parámetros se buscaran en los siguientes intervalos:

Condición Inicial	Cota inferior	Cota superior
$\hat{\sigma}_0$	0.81	0.89
$\hat{\eta}_0$	1.7	2.2
Parámetro	Cota inferior	Cota superior
$\hat{\beta}_s$	0.01	0.03
\hat{K}_s	0.75	1.8
$\hat{\gamma}_s$	0	1
$\hat{\mu}$	0	1

Y se optimizó la función de error con pesos siguientes:

$$\phi(\theta) = \sum_{i=1}^{19} \frac{(\text{Datos}_i - \eta(\hat{\theta}, i))^2}{\text{Datos}_i}$$

Donde $\hat{\theta} = (\hat{\beta}_s, \hat{K}_s, \hat{\gamma}_s, \hat{\mu})$.

El último término se debe a que R_0 es el número básico reproductivo y no puede ser demasiado grande por razones naturales. Por esto mismo se decidió agregar este último término de penalización.

Se hicieron 2 tipos de estimaciones, una con los datos sin perturbaciones obtenidos de los métodos de interpolación 'pchip', y una perturbando los datos con un factor aleatorio con función de distribución $1 + N(0, 0.01)$.

Figura 3.1: Estimación con datos sin perturbación

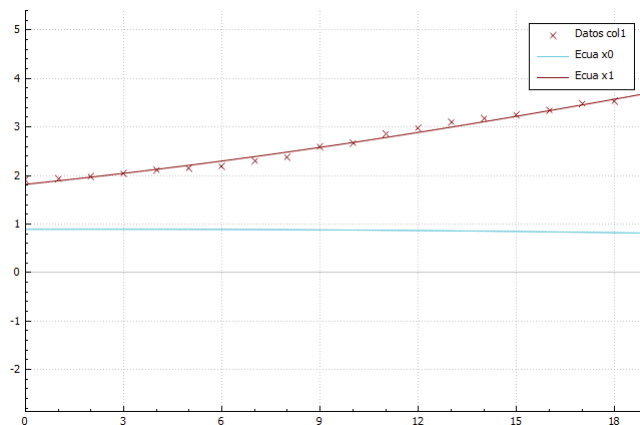
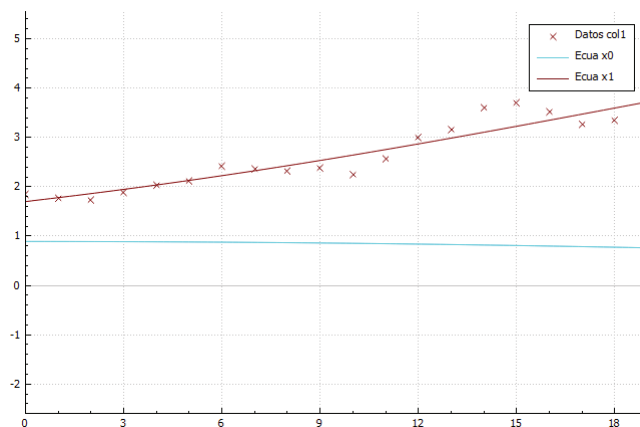


Figura 3.2: Estimación con datos con perturbación:



Los parámetros obtenidos son:

Condición inicial	Sin perturbación	Con perturbación
$\hat{\sigma}_0$	0.89	0.89
$\hat{\eta}_0$	1.81749	1.7
Parámetro	Sin perturbación	Con perturbación
$\hat{\beta}_s$	0.03	0.03
\hat{K}_s	1.8	1.8
$\hat{\gamma}_s$	0.0765177	0.0918295
$\hat{\mu}$	0.0290674	0.0364446

Los errores respectivos de la aproximación son:

	Sin perturbación	Con perturbación
Error relativo	0.0342066	0.324269
Error absoluto	0.108791	0.497871
Error máximo	1.093	3.41786

El número básico reproductivo:

	Sin perturbación	Con perturbación
R_0	4.738	4.535

Tabla de datos obtenidos:

	dat Cper	dat Sper	Apr Sper	Apr Cper
0	1.845540	1.9100	1.81749	1.81749
1	1.766160	1.9327	1.88991	1.90155
2	1.730936	1.9800	1.9653	1.98912
3	1.884558	2.0410	2.04372	2.08022
4	2.031026	2.1110	2.12519	2.17482
5	2.117163	2.1491	2.20974	2.27286
6	2.417187	2.1886	2.29739	2.37429
7	2.359785	2.3000	2.38813	2.479
8	2.319008	2.3747	2.48195	2.58685
9	2.379735	2.5919	2.57881	2.69769

	dat Cper	dat Sper	Apr Sper	Apr Cper
10	2.246063	2.6661	2.67866	2.81131
11	2.565719	2.8579	2.78144	2.92749
12	2.999199	2.9748	2.88705	3.04596
13	3.158342	3.1000	2.99538	3.16642
14	3.602704	3.1757	3.10631	3.28853
15	3.701589	3.2487	3.21967	3.41192
16	3.518640	3.3400	3.33529	3.53621
17	3.265274	3.4810	3.45296	3.66095
18	3.347324	3.5304	3.57247	3.78572
19	3.644591	3.6000	3.69356	3.91002

Donde "dat Cper" = "datos con perturbación", "dat Sper" = "datos sin perturbación", "Apr Sper" = "aproximación de datos sin perturbación", "Apr Cper" = "aproximación de datos con perturbación".

Agradecimientos

Bibliografía

- [ALR] Alavez-Ramírez, J., Estimación de Parámetros en Ecuaciones Diferenciales Ordinarias: Identificabilidad y Aplicaciones a Medicina. PhD thesis, Universidad Nacional Autónoma de México, (2007)
- [ARN] Arnold, V.1: Ordinary differential Equations. Berlin, Heidelberg: Springer-Verlag (1992).
- [BCK] Thomas Back: Evolutionary Algorithms in theory and Practice. Oxford University Press, New York(1996).
- [BRD] Bard, Y.: Nonlinear Parameter Estimation. New York, London: Academic Press (1974).
- [DTT] Datta, Biswa Nath: Numerical Lineal Algebra and Applications 2nd ed. Northern Illinois University Dekalb, Illinois.
- [ESM] Eiben A. E , Smith J.E: Introduction to Evolutionary Computing. Berlin, Heidelberg: Springer-Verlag (2003).
- [KSB] Kowalik J. , Osborne M. R.: Methods for Unconstrained Optimization Problems. American Elsevier Publishing, New York (1968)
- [MRS] Jerrold E. Marsden, Anthony J. Tromva: Vector Calculus 5 fifth ed. W. H Freeman and Company, New York (2003)
- [GNT] Günter Rudolph : Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon (1998), Universitat Dortmund Fachbereich Informatik XI D Dortmund.

