Optimización del Método de Suavizamiento Exponencial con Algoritmo Genético*

Camila Cedeño[1 [0706620713]], Joselyn Chérrez[2 [1726721770]], Gustavo Contreras[3 [0922891817]], Jonathan Guerra[4 [1720064169]], and Alexis Vallejo[5 [1719379842]]

Universidad Politécnica Salesiana, Ecuador

Resumen Este proyecto tiene como finalidad optimizar el alfa, mediante un algoritmo genético, que ayude a realizar dicha acción. ¿Pero que es un algoritmo genético?
El algoritmo genético es un método de optimización basado en la búsqueda partiendo desde los principios de la selección genética y natural; es decir, estos algoritmos se utilizan normalmente para encontrar la mejor solución a problemas complejos del mundo real.

Los dataset utilizados para este proyecto fueron 3:

- **Behavior of the urban traffic of the city of Sao Paulo in Brazil Data Set**: Es un dataset que cuenta con 135 instancias y 18 atributos en el cual se toman muestras cada hora sobre el tráfico y se va a predecir la variable **Slowness in traffic** (la circulación del tráfico) con suavizamiento exponencial y el algoritmo genético.
- **Hungarian Chickenpox Cases Data Set**: El dataset cuenta 521 instancias y 20 atributos y se trata sobre los casos de varicela, se toman muestras cada 7 días desde el año 2005 hasta el 2014 y se va a predecir los casos en la ciudad de **BUDAPEST** con suavizamiento exponencial y el algoritmo genético.
- **Air Quality Data Set**: El dataset cuenta con 9358 instancias y 15 variables y trata de la calidad del aire donde se toman diferentes muestras desde el 10/03/2004 a las 18:00 horas hasta las 14:00 del 04/04/2005 donde se va a predecir la variable **PT08.S5(O3)** respuesta media horaria del sensor (nominalmente dirigido a O3) con suavizamiento exponencial y el algoritmo genético.

Palabras Clave: Población · Cromosomas · Gen · Alelo · Función de aptitud · Genotipo · Fenotipo · Suavizamiento exponencial · Algoritmo genético.

1. Introducción

Se va a optimizar el método de suavizamiento exponencial para series temporales, utilizando los tres datasets aprobados, mediante el algoritmo metaheurístico genético, donde se medirá el tiempo que se demora en ejecutar el algoritmo

* Universidad Politécnica Salesiana

para obtener el valor óptimo de alpha para un CME más bajo y así evaluar el costo computacional dependiendo de los datasets y los diferentes tamaños del dataset ya que existe un pequeño, mediano y grande. Además de las diferentes áreas de estudio de cada uno.

1.1. Suavizado exponencial simple

Es una técnica útil para suavizar una serie de tiempo; proporciona una impresión de los movimientos o patrones generales de los datos. Además, puede utilizarse para obtener predicciones a corto plazo esta técnica supera al método de promedios móviles en el sentido de que no elimina valores. Además, les da mayor ponderación a los últimos valores de la serie y menos peso a los primeros.

Esta herramienta de análisis predice un valor basándose en el pronóstico del período anterior. El suavizado exponencial simple es apropiado cuando las series a predecir son no estacionales y no tienen una tendencia constante ni ascendente ni descendente.

1.2. Suavizado exponencial de Holt

Es una extensión del planteamiento de suavizado exponencial, la diferencia radica en que el procedimiento de suavizado exponencial proporciona una visión de los movimientos a largo plazo sin tener en cuenta la estacionalidad ni la tendencia, mientras que, el Holt permite pronosticar la tendencia. Para utilizar el método de Holt con la finalidad de predcecir, suponemos que todos los movimientos de tendencia futuros continuarán a partir del último nivel suavizado.

1.3. Suavizado exponencial de Winter

Modela el nivel general de la serie, la tendencia y la estacionalidad, por ello se requiere estimar el componente estacional por medio de lo que se conoce como índices estacionales, los cuales sirven para ajustar el modelo[1].

1.4. Algoritmo Genético

Un algoritmo genético es un método de búsqueda que imita la teoría de la evolución biológica de Darwin para la resolución de problemas. Para lo cual, se parte de una población inicial donde se seleccionan los individuos más capacitados para luego reproducirlos y mutarlos para finalmente obtener la siguiente generación de individuos que estarán más adaptados que la anterior generación.[2].

1.5. Terminologías alineadas con Algoritmo Genético

Población Este es un subconjunto de todas las posibles soluciones al problema en cuestión.

Cromosomas Una de las soluciones de la población.

Gen Elemento en el cromosoma.

Alelo El valor numérico asignado a un gen en un cromosoma particular.

Función de aptitud Esta es una función que mejora una salida mediante el uso de una entrada específica. La solución se proporciona como entrada y la compatibilidad de la solución es la salida.

Operaciones genéticas Las mejores personas se casan para tener hijos mejores que sus padres. La composición genética de la siguiente generación se modifica utilizando operadores genéticos.

Genotipo En el espacio de cómputo, la población se conoce como genotipo. Las soluciones se representan en el espacio de cómputo de una manera que un sistema de cómputo puede entender y manipular fácilmente.

Fenotipo La población en el espacio de soluciones del mundo real en el que las soluciones se representan tal como son en entornos del mundo real se denomina fenotipo.

Decodificación y codificación La decodificación es la transformación de una solución del espacio del genotipo al fenotipo, mientras que la codificación es la transformación del espacio del fenotipo al genotipo[3].

2. Materiales y métodos

2.1. Materiales

Hardware
A continuación se presentan las características de la computadora utilizada durante toda la experimentación, para acotar, cabe mencionar que se usó una laptop modelo ASUS N550JK.

Cuadro 1. Características del computador

	Característica	Descripción
CPU	Modelo	Intel Core i7-4720HQ
	Núcleos	12
	Velicidad	2.60 GHz
RAM	Capacidad	18 GB

Software
El Software de simulación utilizado ha sido el R Studio, bajo el lenguaje de programación R.

2.2. Métodos

Limpieza de datos

- Paso 1: Se cargaron los datasets y se verificaron si existían datos faltantes. Siendo este el caso, se procedió a reemplazar por la media de las variables en los datos faltantes para los 3 datasets. Entonces realizando un codigo que indique si existen datos faltantes en los dataset, se confirma que no todos los datasets están con la información completa.
- Paso 2: Se realiza una función para evaluar la normalidad de todas las columnas de los 3 datasets (Figura 1), mediante la prueba de Kolmogorov Smirnov. Y se puso concluir que:
  - Para el primer dataset la variable con mas normalidad fue Slowness in traffic, ya que su p-value (0.00492059908350963) fue el que tuvo el mayor valor con respecto a las otras variables, y es por este motivo que se tomó dicha variable.
  - Para el segundo dataset la variable con mas normalidad fue Budapest, ya que su p-value (3.689629612142e-12) fue el que tuvo el mayor valor con respecto a las otras variables, y es por este motivo que se tomó dicha variable.
  - Para el segundo dataset la variable con mas normalidad fue PT08-S5(O3), ya que su p-value (1.96679782451628e-45) fue el que tuvo el mayor valor con respecto a las otras variables, y es por este motivo que se tomó dicha variable.

```
var_normal <- function(x){
  kolg_srmi <- cbind()
  for( i in colnames(x)){
    k_test <- lillie.test(x[[i]])$p.value
    kolg_srmi <- cbind(kolg_srmi,c(k_test))
  }
  max_Kt=kolg_srmi[which.max(kolg_srmi)]
  indMax_Kt = which.max(kolg_srmi)
  column_nom <- colnames(x[indMax_Kt])
  print(paste("p-value max = ",max_Kt))
  print(paste("Índice de columna: ",indMax_Kt))
  print(paste("Nombre de columna: ",column_nom))
}
```

Figura 1. Resultado de la elección de variables

A continuación en la siguiente tabla (Tabla 2) se muestra a forma de resumen lo antes expuesto en el Paso 2, referente a la comprobación de normalidad.

Cuadro 2. Elección de la columna de cada DataSet

DATASET	COLUMNA ELEGIDA	P-VALUE
1	Slowness in Traffic	0.004920599
2	Budapest	$3.68962961 \times 10^{-12}$
3	PT08S5(O3)	$1.9666797824 \times 10^{-45}$

Diagrama de flujo

La secuencia del diagrama de flujo es la siguiente, primero se crea la población inicial, luego se realiza la evaluación del cromosoma, después la selección del mejor cromosoma, y se realiza el cruce para obtener una mutación, luego se realiza una evaluación de la población, y se entra en un bucle si se obtiene el MSE mínimo se procede a obtener el alfa óptimo para el CME mínimo, y si no se vuelve realizar el mismo proceso hasta obtener el MSE mínimo y obtener el alpha.

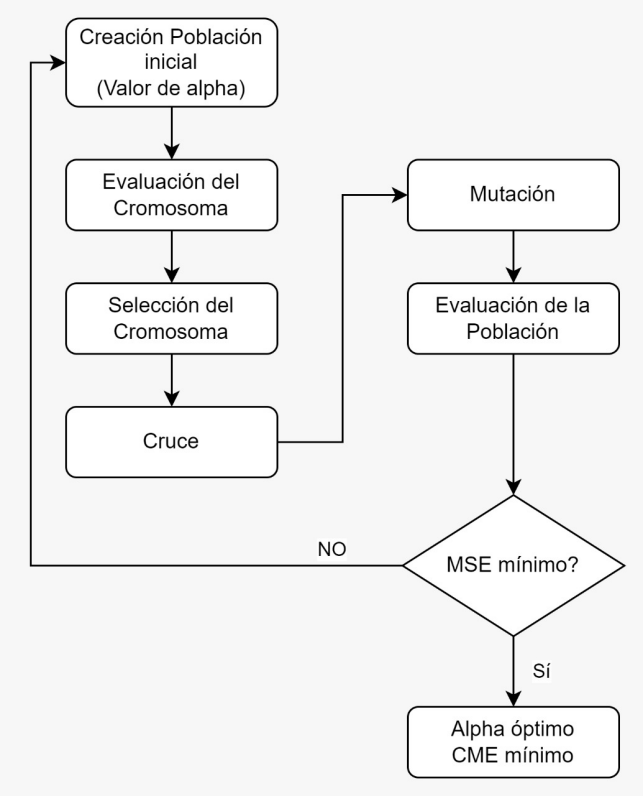

Figura 2. Diagrama de flujo

Diagrama de bloques
El diagrama de bloque consta de lo siguiente:

- En primera instancia se carga el dataset escogido.
- Se hace la predicción mediante el método de zuavizamiento exponencial e introduciéndolo en el algoritmo genético.
- Finalmente se calcula lo errores de predicción como: CME, RMSE, MAE y MASE.

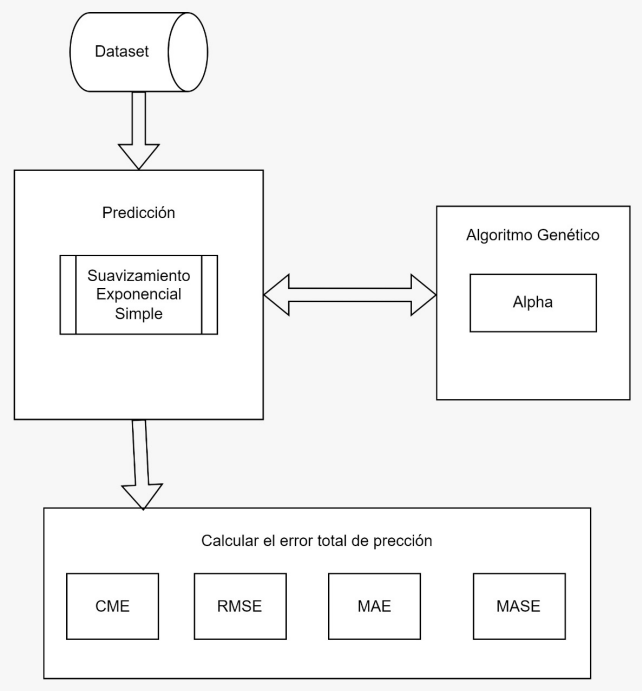

Figura 3. Diagrama de bloques

3. Experimentos

En la presente sección se detallarán cada uno de los resultados referentes al DATASET respectivo, utilizado durante la realización del proyecto. Cabe mencionar que a modo de ejemplificar se detallarán los experimentos uno y dos, en base a los algoritmos utilizados, los cuales fueron:

- Fuerza Bruta
- Algoritmo Genético: Método 1
- Algoritmo Genético: Método 2

3.1. Experimento 1: En el vector de pruebas

Algoritmo de fuerza bruta

A continuación se presentarán los resultados, se puede apreciar que se utilizo el dataset y la función otorgadas de fuerza bruta donde los valores de Y son los siguientes : y = c(17,21,19,23,18,16,20,18,22,20,15,22) entonces se ejecuto la función con saltos de 0.1 y se obtuvo un aplha óptimo de 0.17, además también se obtuvo los valores de MSE, MAE , RMSE y MASE lo cual se demoro en ejecutar un tiempo de 0.06 sec.

```
[1] "Saltos de = 0.01"
[1] "Alpha Optimo = 0.17"
[1] "MSE Minimo = 8.96062474006957"
[1] "MAE = 3.709082437463"
[1] "RMSE = 4.01531802902107"
[1] "MASE = 1.00245471282784"
[1] "Tiempo de Ejecucion:"
0.06 sec elapsed
```

Figura 4. Resultado con algoritmo de fuerza bruta

Algoritmo Genético: Método 1

En la siguiente imagen se ejecuto el algoritmo genético, para el mismo vector de prueba y se obtuvo un valor de alpha de 0.1680 además también se obtuvo los valores de MSE, MAE , RMSE y MASE lo cual se demoro un tiempo de ejecución de 0.2590 secs.

```
-----------------------
Optimización finalizada
-----------------------
Duración selección  = Time difference of 0.259022 secs
Número generaciones = 11
Límite inferior     = 0
Límite superior     = 1
Optimización        = minimizar
valor alpha Óptimo     = 0.1680415
MSE Óptimo encontrado   = 8.959887
MAE     = 2.581814
MASE     = 0.6977874
RMSE     = 2.993567
```

Figura 5. Resultado con algoritmo genético: Método 1

Algoritmo Genético: Método 2

En el metodo 2 se utiliza una libreria establecida en R para algoritmo genetico llamda GA, con este metodo se obtiene un CME de 8.95987 en un tiempo de ejecucion de 0.03 sec.

```
> tic()
> GA <- ga(type="real-valued", fitness = funcion , l(
GA | iter = 1 | Mean = 8.959887 | Best = 8.959887
> print(paste("CME Óptimo",GA@fitnessvalue))
[1] "CME Óptimo 8.95988720167204"
> toc()
0.03 sec elapsed
```

Figura 6. Resultado con algoritmo genético: Método 2

3.2. Experimento 2: Primer Dataset

Algoritmo de fuerza bruta

La tolerancia que se le puso al algoritmo fue de 0.01.
El alpha óptimo encontrado fue: 0.71
Valores de los errores:

- MSE: 1926.4250
- MAE: 25.6411
- RMSE: 45.0636
- MASE: 1.0009

Y su tiempo de ejecución fue de 0.03 segundos.
Ejecutando el algoritmo de fuerza bruta se puede apreciar que no se demora casi nada en ejecutar el algoritmo, debido a su cantidad de instancias del dataset las cuales son 135.

```
[1] "Saltos de = 0.01"
[1] "Alpha Optimo = 0.71"
[1] "MSE Minimo = 1926.42500651869"
[1] "MAE = 25.6411652317369"
[1] "RMSE = 45.0636314404308"
[1] "MASE = 1.0009612491403"
[1] "Tiempo de Ejecucion:"
0.03 sec elapsed
```

Figura 7. Resultado con algoritmo de fuerza bruta

Algoritmo genético: Método 1

El numero de generaciones fue 11.
El límite inferior fue 0.
El límite superior fue 1.
El método de optimización fue minimizar.
El alpha óptimo encontrado fue 0.7087271. Valores de los errores:

- MSE: 1926.407
- MAE: 27.28898
- RMSE: 43.89123
- MASE: 1.065287

Y su tiempo de ejecución fue de 0.2679482 segundos.
Ejecutando el algoritmo de genético con el método 1 se puede apreciar que no se demora casi nada en ejecutar el algoritmo.

```
----------------------
Optimización finalizada
----------------------
Duración selección  = Time difference of 0.2679482 secs
Número generaciones = 11
Límite inferior     = 0
Límite superior     = 1
Optimización        = minimizar
Valor alpha Óptimo     = 0.7087271
MSE Óptimo encontrado   = 1926.407
MAE      = 27.28898
MASE     = 1.065287
RMSE     = 43.89123
```

Figura 8. Resultado con algoritmo genético: Método 1

Algoritmo genético: Método 2

Cabe recalcar que se lo llama con el nombre de método 2 ya que se esta usando la librería que tiene la herramienta de trabajo R. La librería es GA.
Su tiempo de ejecución fue 0.03 segundo.
Valor del error:

- MSE: 1926.4077

Comparado el método 2 con el método 1, arroja el mismo error, pero el tiempo de ejecución del método 2 es menor al método 1.

```
> GA <- ga(type="real-valued", fitness = funcion ,
GA | iter = 1 | Mean = 1926.407 | Best = 1926.407
> print(paste("CME Óptimo = ",GA@fitnessvalue))
[1] "CME Óptimo =  1926.40667939797"
> toc()
0.03 sec elapsed
```

Figura 9. Resultado con algoritmo genético: Método 2

A continuación, se presenta un cuadro resumen de cada experimento realizado y su respectivo resultado.

Cuadro 3. Tabla resumen de Experimentos

		alpha óptimo	**MCE minimo**	**MAE**	**RMSE**	**MASE**	**Tiempo (s)**
Experimento 1 y = c(17,21,19, 23,18,16, 20,18,22, 20,15,22)	**Fuerza Bruta**	0.17	8.96062	3.70908	4.01532	1.00245	0.04
	Algoritmo Genético	0.16804	8.95988	2.58181	2.99356	0.69778	0.25902
	Función Ga - R		8.95989				0.03
Experimento 2 Dataset 1	**Fuerza Bruta**	0.71	1926.42500	25.64116	45.06363	1.00096	0.03
	Algoritmo Genético	0.70872	1926.407	27.28898	43.89123	1.065287	0.26794
	Función Ga - R		1926.40667				0.03
Experimento 3 Dataset 2	**Fuerza Bruta**	0.41	2912.32419	42.47501	64.27954	0.99365	0.03
	Algoritmo Genético	0.39990	2912.263	36.61433	53.96825	0.85655	0.26600
	Función Ga - R		2912.26282				0.04
Experimento 4 Dataset 3	**Fuerza Bruta**	0.99000	38311.51981	129.88746	195.73328	1.00351	0.11
	Algoritmo Genético	0.98890	38158.69	129.9359	195.7773	1.003892	2.78690
	Función Ga - R		38158.685712				0.39

4. Conclusiones

- La programación del algoritmo genético cuenta con varias funciones que hacen posible la optimización del alpha a través del valor mínimo del CME. Estas funciones se ejecutan por separado. Y cuando estas ya hayan sido ejecutadas finalmente se ejecutará el algoritmo genético. Y así se obtiene el valor del alpha optimizado en función del menor CME.
- En tiempos de ejecución los algoritmos de fuerza bruta se demoran más y significan un costo computacional mucho mas alto pero es mas precioso al momento de encontrar alpha, con el método del algoritmo genético el costo computacional es más bajo pero no el mejor pero es de gran ayuda para tener una idea de en que rango se encuentra el valor de alpha para un mejor CME.
- Como hemos visto, la principal ventaja de los algoritmos genéticos radica en su sencillez. Se necesita poca información sobre el espacio de búsqueda porque estamos trabajando en un conjunto de soluciones o parámetros sistematizados (hipotéticos o individuales). Se busca una solución mediante la aproximación de población, en lugar de la aproximación punto a punto. Con un control adecuado, podemos mejorar el promedio de la población, obtener nuevos y mejores individuos y, por lo tanto, tener mejores soluciones.
- El uso del algoritmo genético para la optimización en suavizamiento exponencial, representa un reto en el entendimiento de su funcionamiento. Además, lo más importante es definir la función objetivo, es decir en base a qué optimizará el algoritmo genético.

5. Prospectivas

- Se recomienda que para mas información se debe investigar en Google Schoolar, ya que es ahí donde se encuentra la mejor informacion sobre lo que se quiera consular.
- Se recomienda verificar que los valores del dataset obtengan sean normalizados para no tener problemas al momento de realizar el suavizamiento exponencial.
- La optimización del suavizamiento exponencial depende netamente del entendimiento de la función objetivo, es por ello que su definición es sumamente importante, lo que quiere decir que se debe tener claro el atributo de selección para el algortimo genético con el fin de obtener del mismo el individuo óptimo, y así establecer lo que se debe evaluar para determinar el éxito del mismo, para nuestro caso un CME mínimo.

Referencias

1. Delgadillo Ruiz, O., Ramírez-Moreno, P. P., Leos-Rodríguez, J. A., Salas González, J. M., Valdez-Cepeda, R. D. (2016). Pronósticos y series de tiempo de rendimientos de granos básicos en México. Acta universitaria, 26(3), 23-32.

2. Arranz de la Peña, J., Parra Truyol, A. (2007). Algoritmos genéticos. Universidad Carlos III.
3. ISHA GORASHIYA. (2022, April 12). GENETIC ALGORITHMS IN MACHINE LEARNING - ISHA GORASHIYA - Medium. Medium; Medium. https://medium.com/@isha.gorashiya107894/genetic-algorithms-in-machine-learning-a5218c745068
4. Chusyairi, A. (2018, 1 noviembre). Optimization of Exponential Smoothing Method Using Genetic Algorithm to Predict E-Report Service. IEEE Conference Publication | IEEE Xplore. Recuperado 25 de julio de 2022, de https://ieeexplore.ieee.org/abstract/document/8721008
5. Goodwin, P. (2010). The Holt-Winters Approach to Exponential Smoothing: 50 Years Old and Going Strong. The International Journal of Applied Forecasting, Forthcoming.