Resolviendo el problema de regresión multi-salida mediante *Gene Expression Programming*

Oscar Reyes
Dpto. Informática y Análisis Numérico
Universidad de Córdoba
Email: ogreyes@uco.es

Jose M. Moyano
Dpto. Informática y Análisis Numérico
Universidad de Córdoba
Email: jmoyano@uco.es

Jose M. Luna
Dpto. Informática y Análisis Numérico
Universidad de Córdoba
Email: jmluna@uco.es

Sebastián Ventura
Dpto. Informática y Análisis Numérico
Universidad de Córdoba
Email: sventura@uco.es

***Resumen*—En los últimos años, el estudio de problemas donde los ejemplos están asociados a múltiples variables objetivo al mismo tiempo ha ganado un creciente interés en la comunidad científica. En este trabajo se propone un método basado en *Gene Expression Programming* para darle solución al problema de regresión multi-salida. El método propuesto se enfoca en la regresión simbólica, lo cual permite crear un modelo predictivo multi-salida sin tener conocimiento previo de las relaciones existentes en los datos. La codificación de un individuo de la población se realiza mediante un cromosoma de varios genes, donde cada gen codifica una expresión matemática que produce la salida para una variable objetivo, y así cada individuo de la población representa directamente una posible solución al problema. Los operadores usados en el proceso evolutivo permiten la constante creación de nuevo material genético, y algunos de ellos favorecen la detección de las dependencias existentes entre variables objetivo. El estudio experimental realizado en 8 conjuntos de datos muestra los beneficios y efectividad del método propuesto para resolver el problema de regresión multi-salida.**

I. Introducción

En la última década, el estudio de problemas donde cada ejemplo del conjunto de datos está asociado a múltiples variables objetivo (variables de salida) ha tenido un creciente interés en la comunidad de aprendizaje automático, debido principalmente al elevado número de aplicaciones reales que pueden ser modeladas dentro de este paradigma. El objeto de estudio de la regresión multi-salida o *multi-target regression* (MTR), se enfoca en la predicción simultánea de múltiples variables objetivo continuas usando un único conjunto de variables descriptoras (variables de entrada) [1, 2]. MTR está presente en varios dominios de aplicación, tales como en el modelado ecológico [3], procesado de señales [4], y la eficiencia energética [5].

A día de hoy, se han propuesto un número considerable de métodos para resolver el problema MTR, los cuales se pueden categorizar en dos grupos: métodos de transformación de problemas y métodos de adaptación de algoritmos [1]. Los métodos de transformación descomponen un problema de regresión multi-salida en varios problemas de regresión simple (es decir, con una sola variable objetivo), y por último mediante un proceso de agregación se obtienen las predicciones finales. En este sentido, la mayoría de los métodos de transformación para MTR han sido inspirados en aproximaciones existentes para el aprendizaje multi-etiqueta [2, 6]; en este último paradigma cada ejemplo está asociado a múltiples variables objetivo binarias. Por otro lado, la categoría de adaptación de algoritmos incluye aquellos métodos que no descomponen el problema MTR en problemas de regresión simple, sino que tratan directamente los datos multi-salida. Dentro de esta categoría se han propuesto un amplio número de métodos, tales como métodos estadísticos [7], árboles de regresión [3], algoritmos basados en reglas [8], máquinas de vectores soporte [9] y métodos locales [10].

Entre los paradigmas existentes más populares para resolver problemas de regresión se encuentra la regresión simbólica, donde el objetivo es buscar una expresión matemática que se ajuste a un conjunto de datos dado. En este tipo de técnica no se proporciona ningún modelo en particular como punto de partida, sino que las expresiones se forman por medio de la codificación de bloques matemáticos, lo cual provee de una selección de características implícita en el proceso de construcción de expresiones, y además puede beneficiar la detección de relaciones complejas en los datos. Las técnicas de regresión simbólica han sido aplicadas a un gran número de problemas reales [11], destacando el uso de algoritmos evolutivos para su resolución [12]. Entre las técnicas de computación evolutiva más adecuadas para el desarrollo de métodos de regresión simbólica se encuentra *Gene Expression Programming* (GEP) [12–14], una aproximación que se sitúa entre los algoritmos genéticos y la programación genética, aprovechando así las ventajas de ambas técnicas. [13]. GEP emplea una población de individuos, selecciona padres de acuerdo a su *fitness* y evoluciona la población utilizando operadores genéticos, tal y como hacen los algoritmos genéticos y la programación genética. Sin embargo, la principal diferencia radica en la forma en que los individuos son codificados. En este caso, los individuos se codifican como cadenas lineales de longitud fija (como en algoritmos genéticos), que luego se expresan como árboles de diferentes tamaños y formas (como en programación genética) [13].

Hasta el momento, los trabajos existentes que aplican GEP para regresión simbólica se han restringido a solucionar problemas de regresión simple. Es de destacar que al tratar de solucionar el problema MTR mediante regresión simbólica usando la técnica GEP aparecen dificultades adicionales que no existían en el contexto de la regresión simple, como es la dependencia estadística que puede existir entre las variables objetivo. En este sentido, varios estudios recientes han demostrado que es vital detectar y explotar correctamente dichas dependencias de cara a mejorar el rendimiento predictivo de los algoritmos de regresión [2, 9, 10, 15].

En este trabajo, se propone un método basado en GEP que resuelve el problema MTR mediante regresión simbólica. El método diseñado no divide el problema multi-salida en varias tareas de salida simple, sino que trata directamente con los datos multi-salida, por lo que se obtiene un método con un coste computacional aceptable en conjuntos de datos con un elevado número de variables objetivo. Los individuos se representan mediante un cromosoma con codificación multigénica, donde cada gen representa una función matemática que predice el valor de una variable objetivo; de esta manera cada individuo representa una solución completa al problema MTR. El poder creativo de GEP permite la constante creación de nuevo material genético, permitiendo una mejor exploración del espacio de búsqueda. Además, el método puede detectar las dependencias existentes entre variables objetivo por medio de los operadores de transposición o recombinación de genes entre cromosomas.

La contribución principal de este trabajo es la introducción de un método basado en GEP para el problema MTR. Este trabajo representa un estudio inicial, donde se pretende analizar los beneficios del paradigma GEP para construir modelos de regresión multi-salida. La efectividad del método desarrollado se comprobó mediante un estudio experimental en 8 conjuntos de datos, demostrando que se obtienen resultados prometedores mediante la comparación contra otros dos métodos del estado del arte.

El resto del este trabajo se organiza de la siguiente manera: en la Sección II se describen los fundamentos del método propuesto; la configuración del estudio experimental, así como la discusión de los resultados son presentados en la Sección III; finalmente, en la Sección IV se presentan las conclusiones del presente trabajo.

II. Método Basado en GEP

En esta sección se describe el método propuesto basado en GEP para resolver el problema MTR mediante regresión simbólica. Antes de describir el método en sí, se definen algunas notaciones que son utilizadas a lo largo del trabajo.

Sea $S = \{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), \ldots, (\mathbf{x}^n, \mathbf{y}^n)\}$ un conjunto de datos de n ejemplos de entrenamiento. Una instancia $i \in S$ se representa como una tupla $(\mathbf{x}^i, \mathbf{y}^i)$, donde $\mathbf{x}^i \in \mathcal{X}$ e $\mathbf{y}^i \in \mathcal{Y}$ son los vectores de entrada y salida de i, respectivamente. $\mathcal{X}$ representa el espacio de entrada que contiene d variables descriptivas $x_1, x_2 \ldots, x_d$, mientras que $\mathcal{Y}$ representa el espacio de salida, que contiene q variables objetivo $y_1, y_2 \ldots, y_q$. Por otro lado, x_ℓ^i denota el valor de la ℓ-ésima variable descriptiva para el ejemplo i, mientras que y_ℓ^i representa el valor de su ℓ-ésima variable objetivo. En el problema MTR el objetivo principal es construir una función $f : \mathcal{X} \rightarrow \mathcal{Y}$ que prediga un vector de variables objetivo $\hat{\mathbf{y}}$ a partir de un vector de variables descriptivas nunca antes visto $\mathbf{x}$. Por otro lado, como se hace uso de regresión simbólica, es necesaria la definición del conjunto de símbolos no terminales (funciones) F=$\{-, +, /, *, \ldots\}$ y el conjunto de símbolos terminales T=$\{x_1, x_2, \ldots, x_d\}$, donde cada símbolo terminal representa una variable descriptiva $x_\ell \in \mathcal{X}$.

En el paradigma GEP, un gen está compuesto por una cabeza y una cola. La cabeza puede contener símbolos que pertenezcan tanto a los conjuntos F como T, mientras que la cola puede contener solo símbolos terminales. La longitud de la cabeza (h) se escoge de acuerdo al problema, mientras que la longitud de la cola (t) se calcula como t=$h(a-1)+1$, donde a es el número de argumentos de la función con máxima aridad en F. Tenga en cuenta que a mayor h, más larga y compleja será la expresión matemática que se podrá codificar, sin embargo hay que considerar que también será mayor el espacio de búsqueda. Una desventaja de la mayoría de métodos basados en GEP reside en la continua transformación de los cromosomas en árboles de expresión (ETs, por sus siglas en inglés) y su correspondiente evaluación para obtener el *fitness* de los individuos. Para evitar esto, en este trabajo los genes se codifican mediante notación prefija, siguiendo el enfoque propuesto por Peng et al. en [12], permitiendo la evaluación de los individuos sin construir los ETs, y mejorando por tanto significativamente la eficiencia computacional de GEP. La Figura 1 representa un gen de un cromosoma, que en este caso contiene la notación prefija de la ecuación matemática $f(\mathbf{x}) = \sqrt{\left(\frac{x_1}{x_4 * x_5} - \sqrt{x_1}\right)} + x_1$.

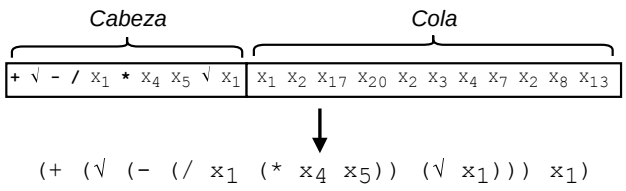

Figura 1. Un gen con $h = 10$ y su correspondiente fenotipo representado en notación prefija.

Se define una población de p individuos, donde cada individuo representa una posible solución completa al problema MTR. Para ello, un individuo es representado mediante un cromosoma multi-génico, compuesto por q genes de igual longitud $c = [g_1, g_1, \ldots, g_q]$, conteniendo un gen por cada variable objetivo del problema. Por tanto, el gen ℓ-ésimo de un cromosoma codifica una función matemática que se puede utilizar posteriormente como modelo predictivo para estimar el valor de la variable objetivo y_ℓ^i de un ejemplo de prueba i.

Para calcular el nivel de adaptación de un individuo se calcula el error cuadrático medio relativo (*average relative*

root mean square error, aRRMSE) en promedio para todas las variables objetivo sobre el conjunto de entrenamiento S,

$$\frac{1}{q}\sum_{\ell=1}^{q}\sqrt{\frac{\sum_{i\in S}(y_\ell^i-\hat{y}_\ell^i)^2}{\sum_{i\in S}(y_\ell^i-\bar{y}_\ell)^2}}\,, \tag{1}$$

donde y_ℓ^i e $\hat{y}_\ell^i$ son los valores de la ℓ-ésima variable objetivo en los vectores de salida conocido ($\mathbf{y}^i$) y predicho ($\hat{\mathbf{y}}^i$) para el ejemplo i, respectivamente. Por otro lado, $\bar{y}_\ell$ es el valor medio de la variable objetivo ℓ-ésima en el conjunto de entrenamiento S. Resumiendo, dado un individuo de la población, la función de *fitness* mide el error cuadrático medio relativo (RRMSE) para la primera variable objetivo, evaluando la función matemática codificada en el primer gen del cromosoma en cada ejemplo de entrenamiento $i \in S$, y así este proceso se realiza para todas las variables objetivo obteniendo finalmente el valor promedio entre todas ellas.

La población inicial se crea de manera aleatoria, pero siempre controlando la diversidad entre los individuos generados. Un cromosoma se forma creando cada uno de sus genes de la siguiente manera: (I) los símbolos en la cabeza del gen se seleccionan aleatoriamente de $F \cup T$, asegurando que el primer elemento (raíz) de la cabeza sea un símbolo no terminal; y (II) los símbolos de la cola se seleccionan aleatoriamente del conjunto T. Un individuo se añade a la población inicial si tiene una similaridad respecto al resto de la población menor que un umbral específico. La similaridad entre dos individuos se calcula como

$$s(c^i, c^j) = 1 - \frac{\sum_{\ell=1}^{q} h(c_\ell^i, c_\ell^j)}{q^2(t+h)}, \tag{2}$$

donde c^i y c^j son los cromosomas de los individuos i y j, respectivamente. La función $h(c_\ell^i, c_\ell^j)$ mide el número de símbolos distintos, sumados a lo largo de las $t+h$ posiciones, en el ℓ-ésimo gen de los cromosomas c^i y c^j. Este método permite la creación de una población inicial con suficiente diversidad, lo cual evita la convergencia temprana del método a un mínimo local.

Respecto al operador de selección, se ha utilizado una selección por torneo binario para crear la población intermedia de padres. Se empleó una baja presión selectiva para que no solo se favorezca a los mejores individuos en el proceso de selección, sino que se estimule también la diversidad entre los individuos seleccionados.

En cuanto al operador de mutación, los padres mutan con una tasa de mutación p_m y se ha diseñado un operador que siempre genera individuos válidos. Se han empleado varios puntos de mutación por cromosoma (n_{mp}), y dicho operador realiza los siguientes pasos para cada uno de los puntos de mutación: (I) se selecciona aleatoriamente uno de los genes del individuo; (II) se selecciona aleatoriamente una posición dentro del gen; (III) si la posición pertenece a la cabeza, el símbolo en dicha posición se cambia aleatoriamente por un símbolo que pertenece a $F \cup T$, excepto en el caso de que la posición seleccionada sea la raíz, donde se reemplazará necesariamente por un símbolo en F; (IV) si la posición pertenece a la cola, el símbolo en la posición seleccionada se cambia por otro símbolo en T.

Una característica de GEP que lo diferencia de otros paradigmas evolutivos es que fragmentos del genoma se pueden activar y saltar a otro lugar en el cromosoma (elementos transponibles), y esta característica da lugar a la definición de los siguientes operadores de transposición: *insertion sequence* (IS), *root insertion sequence* (RIS) y *gene transposition* (GT). El operador IS se utiliza con una pequeña probabilidad p_{is}, y permite que pequeños fragmentos de un gen (fragmentos con una función o un terminal en la primera posición) se transpongan a la cabeza de los genes, excepto a la raíz; este operador se ejecuta n_{is} veces sobre un cromosoma. El operador IS realiza los siguientes pasos: (I) dado un cromosoma, se selecciona un gen a partir del cual un fragmento de longitud entre $[1, l_{is}]$ (l_{is}-máxima longitud de un fragmento) se selecciona aleatoriamente; y (II) se selecciona un gen de destino donde insertar el fragmento copiado empezando en una posición aleatoria de la cabeza del gen (distinta de la raíz).

Por otra parte, el operador RIS se usa con una probabilidad baja p_{ris}, y permite que fragmentos pequeños con un símbolo no terminal en la primera posición se transpongan a la raíz de los genes; este operador se realiza n_{ris} veces sobre un cromosoma. Para ello, este operador realiza los siguientes pasos: (I) dado un cromosoma, se selecciona aleatoriamente un gen; (II) se selecciona un punto aleatorio en la cabeza del gen seleccionado y se busca hacia atrás hasta encontrar un símbolo no terminal; (III) y este fragmento se copia empezando en la raíz del gen. Durante la transposición RIS, los símbolos en la cabeza se desplazan hacia la derecha hasta acomodar completamente el nuevo fragmento copiado. Al igual que en el operador de transposición IS, la cola del gen que se transpone y los genes adyacentes no se modifican.

Por último, el operador GT se usa con una pequeña probabilidad p_{gt}, y permite que genes completos se transpongan al principio de los cromosomas. Para ellos, este operador realiza los siguientes pasos: (I) dado un cromosoma, se selecciona aleatoriamente un gen y se elimina de su lugar de origen; (II) y el gen se copia al principio del cromosoma, mientras que los genes restantes se desplazan a la derecha, manteniéndose la longitud original del cromosoma. Los operadores IS, RIS y GT producen siempre individuos válidos, sin embargo, hay que tener en cuenta que dichos operadores pueden modificar drásticamente la expresión de un gen; cuanto más cercano a la raíz del gen ocurra la transposición, más profundo será el cambio en la expresión del gen [13].

En el método propuesto también se aplicaron tres tipos de recombinación o cruce: recombinación en un punto (OP), recombinación en dos puntos (TP), y recombinación de genes (GP), con probabilidades de aplicación p_{op}, p_{tp}, y p_{gp}, respectivamente. Dados dos cromosomas, el operador OP realiza el cruce sobre un punto seleccionado aleatoriamente. El operador TP seleccionan aleatoriamente dos puntos y los fragmentos entre dichos puntos se intercambian entre ambos padres. En GP se intercambian los genes, ubicados en una

posición seleccionada aleatoriamente, entre ambos padres. Es de destacar que en los tres operadores, se cruzan dos padres y se obtienen dos nuevos individuos válidos. Además, el operador TP tiene más poder de transformación que el OP en problemas más complejos, especialmente cuando se utilizan cromosomas multigénicos. Por último, el operador GP no crea genes nuevos como OP y TP, sin embargo, cabe destacar que este operador puede introducir nuevo material en la población, ya que los genes intercambiados entre padres pueden ser muy diferentes.

Finalmente, se ha seguido un esquema generacional para actualizar la población que pasa de una generación a otra, donde el peor individuo de la nueva población es reemplazado por el mejor individuo de la población anterior. El método propuesto (GEPMTR, de ahora en adelante) sigue los pasos que se muestran en la Figura 2. Estos pasos se realizan n_g veces (número de generaciones), y al final del proceso se selecciona el mejor individuo encontrado a lo largo de todas las generaciones, cuyo cromosoma representa un modelo de regresión multi-salida que puede ser utilizado posteriormente para predecir las variables objetivo de conjuntos de ejemplos nunca antes visto.

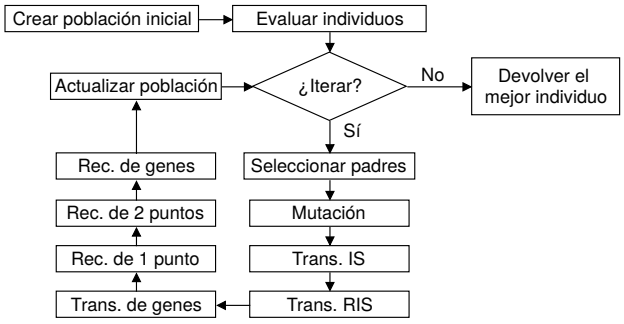

Figura 2. Diagrama de flujo del método GEPMTR.

III. Estudio Experimental

En esta sección primero se explica la configuración de los experimentos realizados en este trabajo, y posteriormente se presenta un análisis y discusión de los resultados obtenidos.

III-A. Configuración de los Experimentos

Los experimentos se han ejecutado sobre ocho conjuntos de datos de regresión multi-salida[1]. La Tabla I muestra los conjuntos de datos utilizados y sus características, como el número de instancias (n), variables descriptivas (d) y variables objetivo (q).

En el estudio experimental, se compara nuestra propuesta (GEPMTR) con otros dos algoritmos del estado del arte para regresión multi-salida, *Single Target* (ST) y *Regressor Chains* (RC) [2]. Tanto ST como RC usan árboles de regresión como predictores de regresión simple, tal y como recomiendan sus autores. Los parámetros utilizados para la ejecución de GEPMTR se muestran en la Tabla II; la mayoría de los valores son los recomendados por Ferreira [13]. Además, se utilizó una probabilidad global de cruce de 0.8, que corresponde con la suma de las probabilidades de los tres operadores de cruce utilizados.

[1]Los conjuntos de datos están disponibles públicamente en http://mulan.sourceforge.net/datasets-mtr.html

Tabla I
Conjuntos de datos de regresión multi-salida.

	n	d	q
slump	103	7	3
jura	359	15	3
sf1	323	10	3
atp1d	337	411	6
osales	639	413	12
wq	1060	16	14
oes97	334	263	16
oes10	403	298	16

Tabla II
Parámetros del algoritmo GEPMTR.

Parámetro	**Valor**
Conjunto de funciones (F)	$\{-, +, /, *, \sqrt{}, sin, cos, log\}$
Conjunto de terminales (T)	$\{x_1, x_2, \ldots, x_d\}$
Tamaño de la población (p)	100
Número de puntos de mutación (n_{mp})	2
Probabilidad de mutación (n_{mp})	0,1
Número de fragmentos IS (n_{is})	3
Máx. longitud de los fragmentos IS (l_{is})	3
Probabilidad IS (p_{is})	0,1
Número de fragmentos RIS (n_{ris})	3
Probabilidad RIS (p_{ris})	0,1
Probabilidad GT (p_{gt})	0,1
Probabilidad OP (p_{op})	0,2
Probabilidad TP (p_{tp})	0,5
Probabilidad GP (p_{gp})	0,1

Para la evaluación de los algoritmos, se utilizó la medida aRRMSE, que ha sido ampliamente usada para la evaluación de métodos de regresión multi-salida [1, 2]. La medida aRRMSE se calcula como se mostró en la Ecuación 1. En todos los experimentos, el valor de aRRMSE se estimó realizando un *10-folds cross-validation*, midiendo el error en cada uno de los conjuntos de test y promediando finalmente los resultados.

Para realizar comparaciones múltiples entre algoritmos, se ha utilizado el test de Friedman [16]. En casos donde el test de Friedman indicase que existen diferencias significativas en el rendimiento de los algoritmos, se ha realizado el test *post-hoc* de Holm [17] para realizar una comparación múltiple con un algoritmo de control. Por otra parte, en los casos donde se compararon dos métodos independientes, se empleó el test de Wilcoxon [18].

III-B. Resultados Experimentales

En el estudio experimental, primero se analizó el comportamiento de GEPMTR considerando diferentes valores del parámetro h= $\{5, 10, 15, 20\}$ y número de generaciones. La Tabla III muestra los resultados, donde en cada fila se han resaltado en negrita los mejores valores para cada h.

Se puede observar como en prácticamente todos los casos, cuando GEPMTR se ejecutó con 500 generaciones, los individuos evolucionaron hacia un mejor modelo predictivo,

Tabla III
Resultados de GEPMTR dependiendo de h y del número de generaciones.

	$h=5$		$h=10$		$h=15$		$h=20$	
	100 g.	**500 g.**	**100 g.**	**500 g.**	**100 g.**	**500 g.**	**100 g.**	**500 g.**
slump	**0,860**	0,872	0,848	**0,772**	0,864	**0,767**	0,845	**0,752**
jura	0,692	**0,669**	0,681	**0,646**	0,686	**0,644**	0,684	**0,642**
sf1	1,129	**1,083**	1,128	**1,128**	**1,142**	1,164	1,156	**1,141**
atp1d	0,555	**0,446**	0,541	**0,453**	0,545	**0,450**	0,544	**0,450**
osales	0,954	**0,860**	0,910	**0,861**	0,908	**0,881**	0,921	**0,888**
wq	0,976	**0,968**	0,974	**0,963**	0,975	**0,961**	0,976	**0,960**
oes97	0,660	**0,589**	0,650	**0,601**	0,658	**0,598**	0,666	**0,623**
oes10	0,530	**0,471**	0,530	**0,520**	0,546	**0,484**	0,541	**0,487**
p-value	**0,023**		**0,022**		**0,030**		**0,014**	

obteniendo un mejor rendimiento. Para cada valor diferente de h, se realizó el test de Wilcoxon para comparar el rendimiento de GEPMTR con 100 y 500 generaciones. Los *p-values* del test de Wilcoxon se muestran en la última fila de la Tabla III. El test estadístico rechazó la hipótesis nula en todos los casos para un nivel de significación α=$0,05$, lo cual demuestra que el algoritmo con 500 generaciones obtiene resultados significativamente mejores que con 100 generaciones.

Una vez verificado que GEPMTR funciona mejor con un mayor número de generaciones, el estudio se centra en el valor de h. La Tabla IV muestra los *rankings* de GEPMTR ejecutado con diferentes valores de h y con 500 generaciones sobre cada conjunto de datos. Esta tabla es un resumen de la Tabla III, pero esta se centra en los valores de *ranking* en lugar de aRRMSE. En cada fila, el algoritmo con un mejor rendimiento obtiene un *ranking* de 1, el siguiente un *ranking* de 2, y así sucesivamente. La última fila muestra el valor de *ranking* medio devuelto por el test de Friedman.

Tabla IV
Valores de ranking con diferentes valores de h y 500 generaciones.

	h=5	h=10	h=15	h=20
slump	4,00	3,00	2,00	**1,00**
jura	4,00	3,00	2,00	**1,00**
sf1	**1,00**	2,00	4,00	3,00
atp1d	**1,00**	4,00	2,50	2,50
osales	**1,00**	2,00	3,00	4,00
wq	4,00	3,00	2,00	**1,00**
oes97	**1,00**	3,00	2,00	4,00
oes10	**1,00**	4,00	2,00	3,00
Ranking medio	**2,13**	3,00	2,44	2,44

Se puede observar que, a pesar de que el test de Friedman no encontró diferencias significativas en el rendimiento de GEPMTR con los diferentes valores del parámetro h, la configuración con h=5 fue la que mejor *ranking* medio obtuvo, mostrando una tendencia a que cuanto más simples sean los modelos, mejor es su rendimiento predictivo.

En la segunda fase del estudio experimental, se realizó una comparación entre GEPMTR y dos algoritmos del estado del arte en MTR. La Tabla V muestra los resultados; para esta comparación, los resultados que se muestran para GEPMTR son el mínimo de las diferentes configuraciones de h con 500 generaciones. La última fila de la tabla muestra los valores de *ranking* medio obtenidos por el test de Friedman. En cada fila, el mejor valor de error se resalta en negrita. Los resultados muestran que el método propuesto obtiene mejores resultados que los otros dos algoritmos en siete de los ocho conjuntos de datos considerados, obteniendo también el mejor valor de *ranking* medio.

Tabla V
Comparación de GEPMTR con dos algoritmos del estado del arte en MTR.

	GEPMTR	**ST**	**RC**
slump	**0,752**	0,814	0,829
jura	**0,642**	0,696	0,704
sf1	1,083	1,127	**1,046**
atp1d	**0,446**	0,479	0,484
osales	**0,860**	0,925	0,965
wq	**0,960**	0,966	0,974
oes97	**0,589**	0,716	0,719
oes10	**0,471**	0,594	0,595
Ranking medio	**1,125**	2,125	2,750

El estadístico del test de Friedman fue igual a $10,75$, y la hipótesis nula fue rechazada con un *p-value* de $0,005$, existiendo diferencias significativas en el rendimiento de los algoritmos. Para detectar dónde estaban localizadas dichas diferencias se ejecutó el test de Holm, cuyos resultados se muestran en la Tabla VI, indicando que GEPMTR tiene un rendimiento predictivo significativamente mejor que el resto de algoritmos para un nivel de significación $\alpha = 0,05$.

Tabla VI
p-values ajustados calculados por el test de Holm.

	ST	**RC**
GEPMTR vs.	0,045	0,002

III-C. Discusión

La representación multi-génica utilizada resultó ser una manera efectiva para representar los cromosomas, permitiendo que cada individuo constituya una solución diferente y completa para el problema MTR. La manera de crear la población inicial, así como el uso de un operador de selección con menor presión, permitió la evolución de una población con suficiente diversidad previniendo que el método quedase atrapado en un mínimo local o en valles aplanados en las primeras generaciones.

Operadores como el de transposición IS, y los de recombinación OP y TP pueden introducir material genético beneficioso. Así, buenos bloques existentes en genes más adaptados se pueden transferir a otros genes, favoreciendo el intercambio de información entre genes, y por tanto, el modelado de relaciones estadísticas entre variables objetivo.

En cuanto a la eficiencia computacional de GEPMTR, el hecho de haber usado notación prefija, y por tanto que los genes puedan ser evaluados sin construir su correspondiente árbol de expresión, hace que la evaluación de los individuos se haya acelerado significativamente. GEPMTR puede ser paralelizado fácilmente, y por tanto, se puede reducir significativamente su

tiempo de cómputo. Por otro lado, no se consume tiempo en la descomposición del problema MTR en varios problemas de regresión simple, sino que el método propuesto trata directamente con los datos multi-salida. Además, GEPMTR tiene implícito un proceso de selección de características que puede ser muy beneficioso en el proceso de aprendizaje de modelos de regresión; la expresión matemática codificada por un gen involucra un número de variables mucho menor que el total de variables descriptivas existentes. Todas las características antes mencionadas permiten la aplicación efectiva del método propuesto en conjuntos de datos de gran escala.

GEPMTR obtuvo resultados prometedores en los experimentos realizados. Sin embargo, la principal desventaja de GEPMTR radica en el análisis de un considerable número de hyperparámetros. Para futuros trabajos será importante expandir el estudio experimental, por ejemplo analizando diferentes valores para las probabilidades de los operadores.

IV. Conclusiones

En este trabajo se ha presentado un método basado en GEP que resuelve el problema MTR mediante la técnica de regresión simbólica. Tanto la forma de codificación y evaluación de los individuos, como la posibilidad de paralelización, resultan en un método con un coste computacional aceptable y que permite ser usado en conjuntos de datos de gran escala.

Este trabajo corresponde con un estudio inicial, donde se busca analizar los beneficios del paradigma GEP para construir modelos de regresión multi-salida. En trabajos futuros, se estima mejorar algunos de los componentes del algoritmo, como un operador de selección que permita ajustar la presión selectiva. Por otro lado, sería interesante el desarrollo de *ensembles* usando GEPMTR.

Agradecimientos

Este trabajo ha sido financiado por el proyecto TIN2017-83445-P del Ministerio de Economía y Competitividad y Fondos FEDER, y además por la ayuda FPU del Ministerio de Educación FPU15/02948.

Referencias

[1] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[2] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas., "Multi-target regression via input space expansion: Treating targets as inputs," *Machine Learning*, vol. 104, no. 1, pp. 55–98, 2016.

[3] D. Kocev, S. Džeroski, M. D. White, G. R. Newell, and P. Griffioen, "Using single and multi-target regression trees and ensembles to model a compound index of vegetation condition," *Ecological Modelling*, vol. 220, no. 8, pp. 1159–1168, 2009.

[4] D. Tuia, J. Verrelst, L. Alonso, F. Pérez-Cruz, and G.Camps-Valls, "Multioutput support vector regression for remote sensing biophysical parameter estimation," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 804–808, 2011.

[5] A. Tsanas and A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools," *Energy and Buildings*, vol. 49, no. 560-567, 2012.

[6] O. Reyes, C. Morell, and S. Ventura, "Evolutionary feature weighting to improve the performance of multi-label lazy algorithms," *Integrated Computer-Aided Engineering*, vol. 21, no. 4, pp. 339–354, 2014.

[7] T. Similä and J. Tikka, "Input selection and shrinkage in multiresponse linear regression," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 406–422, 2007.

[8] T. Aho, S. D. B. Ženko, and T. Elomaa, "Multi-target regression with rule ensembles," *Journal of Machine Learning Research*, vol. 373, pp. 2055–2066, 2009.

[9] G. Melki, A. Cano, V. Kecman, and S. Ventura, "Multi-target support vector regression via correlation regressor chains," *Information Sciences*, vol. 415-416, pp. 53–69, 2017.

[10] O. Reyes, A. Cano, H. Fardoun, and S. Ventura, "A locally weighted learning method based on a data gravitation model for multi-target regression," *International Journal of Computational Intelligence Systems*, vol. 11, pp. 282–295, 2018.

[11] S. Stijven, E. Vladislavleva, A. Kordon, L. Willem, and M. E. Kotanchek, *Genetic Programming Theory and Practice XIII*. Cham: Springer, 2016, ch. Prime-Time: Symbolic Regression Takes Its Place in the Real World, pp. 241–260.

[12] Y. Peng, C. Yuan, X. Qin, J. Huang, and Y. Shi, "An improved gene expression programming approach for symbolic regression problems," *Neurocomputing*, vol. 137, pp. 293–301, 2014.

[13] C. Ferreira, "Gene expression programming: A new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, pp. 87–129, 2001.

[14] H. S. Lopes and W. R. Weinert, "Egipsys: An enhanced gene expression programming approach for symbolic regression problems," *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 3, pp. 375–384, 2004.

[15] J. M. Moyano, E. Gibaja, and S. Ventura, "An evolutionary algorithm for optimizing the target ordering in ensemble of regressor chains," in *IEEE Congress on Evolutionary Computation*, 2017, pp. 2015–2021.

[16] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[17] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, pp. 65–70, 1979.

[18] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.