EL7037 - Tarea Computacional N°1
Programación Genética

Profesor de cátedra: Pablo Estévez
Profesor auxiliar: Jorge Vergara

Semestre Primavera 2025

1 Regresión simbólica de funciones

Se desea obtener mediante programación genética, una expresión matemática que se asemeje a la siguiente función:

$$f(x) = 2 \cdot x^3 - x^2 - x + 5$$

de manera que el error entre las salidas de los individuos y las salidas deseadas en ciertos puntos $(x, f(x))$ sea mínimo. Para ello considere lo siguiente:

1. Cree las salidas deseadas para un determinado conjunto de valores de entrada. Para esto, realice un muestreo equidistante de 50 puntos de la función f(x) y decida el rango x de valores dentro del cual realizará el muestreo.
2. Busque mediante PG el individuo-programa que mejor se adapte a sus datos tomando en cuenta:
   (a) Aplicación de los cinco pasos preparatorios definidos por John Koza (el conjunto de terminales, el conjunto de funciones primitivas, la medida de fitness, los parámetros que controlan la simulación, el método para designar un resultado, y criterio para terminar una simulación)
   (b) Evolución con 1000 generaciones.
3. Realización de nuevas simulaciones combinando distintos valores de:
   (a) Profundidad máxima (PM) de los árboles (PM = 4 y 14)
   (b) Número máximo de generaciones (NMG) de evolución (NMG = 12 y 100)
   (c) Diferentes combinaciones de valores de mutación, crossover y selección (definido por usted)
4. Cree las salidas de dos conjuntos de test para probar la capacidad de generalización de los árboles encontrados con puntos de la función f(x) no usados en la evolución:
   (a) Test 1: 50 valores escogidos aleatoriamente dentro del rango utilizado para el entrenamiento
   (b) Test 2: 50 valores escogidos fuera del rango de entrenamiento

Una vez obtenida las simulaciones analice y grafique lo siguiente:

- Fitness (mejor, peor, promedio de la población).
- Complejidad de los árboles (fitness vs. número de nodos, número de nodos vs. generaciones)
- Rapidez de convergencia
- ¿Se obtuvo la misma solución-función a la deseada? Justifique.

Para lo anterior considerar el promedio de 5 simulaciones con inicialización aleatoria de la población.

2 Proyección de datos

Cualquier objeto puede ser descrito en términos de un conjunto de características pertenecientes a un espacio que usualmente es de gran dimensionalidad. Uno de los objetivos en las técnicas de proyección (visualización) es extraer la mayor cantidad de información de los datos originales pertenecientes a un espacio X ($X|x_i \in \mathbb{R}^D$) para proyectarlos a un espacio de salida Y ($Y|y_i \in \mathbb{R}^d$) de tal forma que éste último sea de menor dimensionalidad que el original ($d << D$). Generalmente se intenta representar los datos en un espacio 2D o 3D con fines de visualización.

A pesar que en la literatura existen muchos algoritmos de proyección, para la mayoría de éstos es imposible proyectar datos que no hayan sido procesados por el algoritmo anteriormente, es decir, no pueden generalizar directamente a nuevos datos. La programación genética ofrece una solución a este problema ya que no solo proyecta los datos (utilizando como función fitness el funcional de uno de estos métodos), sino que además crea relaciones entre las variables permitiendo de esta forma generalizar la proyección a datos nuevos.

El objetivo del siguiente problema es precisamente probar la utilidad de la programación genética en proyección. Para ello deberá proyectar 2 bases de datos a un plano 2D, utilizando como función fitness el funcional de Sammon (ver anexo).

Las consideraciones de las simulaciones son:

1. Crear conjuntos de entrenamiento (60%) y test (40%) obtenidos de cada base de dato (el mejor muestreo debe mantener los cluster existentes en cada conjunto).
2. Aplicación de los cinco pasos preparatorios definidos por John Koza.
3. Evolución con 5000 generaciones como máximo.
4. Profundidad máxima de los árboles (PM = 5,10,15,20)
5. Diferentes combinaciones de valores de mutación, crossover y selección (definido por usted)

Una vez obtenida las simulaciones analice y grafique lo siguiente:

- Fitness (mejor, peor, promedio de la población)
- Complejidad de los árboles (fitness vs. número de nodos, número de nodos vs. generaciones)
- Rapidez de convergencia
- Proyección de los mejores árboles encontrados en cada caso. Justifique este análisis utilizando métricas de proyección (ver anexo).
- En relación a los mejores árboles obtenidos anteriormente, encuentre en las simulaciones anteriores otro árbol de menor complejidad tal que el desempeño de la proyección sea comparable con el mejor.

Para lo anterior considerar el promedio de 5 simulaciones con inicialización aleatoria de la población.

Para realizar la tarea se dispondrá del toolbox de programación genética GPalta, el cual esta desarrollado en JAVA y que es compatible con Python. Además se adjunta las bases de datos utilizadas en el problema 2, junto con un set de métricas utilizadas en proyección (ver anexo).

Se pide entregar un informe escrito que contenga obligatoriamente las siguientes secciones:

1. INTRODUCCIÓN: Describir brevemente el problema a resolver, la estructura del informe y consideraciones generales
2. DESARROLLO: Indicar el procedimiento usado para resolver el problema y los fundamentos teóricos en que se basa este procedimiento
3. RESULTADOS Y GRÁFICOS: Los datos de salida del algoritmo genético deben ser procesados de manera de ilustrar los resultados en tablas y gráficos con fines del informe (No entregar listados de datos)
4. CONCLUSIONES: Realizar una discusión respecto de los resultados, y contrastarlos con la teoría. Agregar comentarios y opiniones personales acerca de cómo mejorar los resultados.
5. ANEXOS: Listado comentado del programa

- Subir a carpeta Tareas en u-cursos con el informe en formato digital (pdf) y el código fuente.
- El informe impreso y digital no deben exceder las 20 páginas.
- No serán considerados en la evaluación del informe gráficos que no sean legibles (colores, tamaño, etc.).
- Plazo de entrega:
  - Informe digital y códigos fuentes: Martes 26 de agosto de 2025 hasta las 23:59:59 horas a través de u-cursos.

ANEXO

Bases de datos

Nombre	Nº muestras	Nº variables	Nº cluster
Atom	800	3	2
Derm	358	34	6

Métricas de proyección

Estas medidas están implementadas en Matlab. Solo se requiere comprenderlas y usarlas

Confiabilidad (trustworthiness)

Confiabilidad se refiere al hecho de que si un conjunto de datos son vecinos en el espacio de salida, también deben serlo en el espacio de entrada.

Sea N un número total de datos y r_{ij} el ranking de distancias del dato j hacia el dato i en el espacio de entrada. Sea $U_k(i)$ el conjunto de aquellos datos que están en una vecindad de tamaño k de la muestra i en el espacio de salida, pero no en el espacio de entrada. La medida de fiabilidad de la proyección es.

$$M_1 = 1 - \frac{2}{N \cdot k \cdot (2N - 3k - 1)} \cdot \sum_{i=1}^{N} \sum_{j \in U_k(i)} (r_{ij} - k)\,.$$

Continuidad

En forma inversa a la medida de fiabilidad esta la medida de continuidad, la cual dice si los datos que son vecinos en el espacio de entrada deben ser vecinos en el espacio proyectado. Sea $V_k(i)$ el conjunto de datos en la vecindad del dato i en el espacio original pero no en la proyección, y sea s_{ij} el ranking del dato j en orden de acuerdo a la distancia desde i en el espacio proyectado. La medida de continuidad de la proyección es medida por

$$M_2 = 1 - \frac{2}{N \cdot k \cdot (2N - 3k - 1)} \cdot \sum_{i=1}^{N} \sum_{j \in V_k(i)} (s_{ij} - k)\,.$$

Conservación de la topología (qm)

Corresponde a una evaluación del orden en que los datos son vecinos en el espacio original y cómo estos se mantienen en espacio proyectado. Los n vecinos más cercanos NN_{ij} $(i \in [1, n],\ j \in [1, N]$ de cada vector v_j son computada. La medida de topología (qm) es calculada como

$$q_m = \frac{1}{3 \cdot n \cdot N} \sum_{j=1}^{N} \sum_{i=1}^{n} q_m(i, j),$$

donde

$$q_m(i,j) = \begin{cases} 3, & si\ NN_{ji}^X = NN_{ji}^Y \\ 2, & si\ NN_{ji}^X = NN_{jl}^Y,\ l \in [1, n],\ i \neq l \\ 1, & si\ NN_{ji}^X = NN_{jt}^Y,\ t \in [n, k],\ n < k \\ 0, & otro\ caso \end{cases}$$

NN_{ij}^X corresponde a la distancia entre los datos i y j en el espacio de entrada (espacio original) X.
NN_{ij}^Y corresponde a la distancia entre los datos i y j en el espacio de entrada (espacio proyectado) Y.