



Universidad Simón Bolívar

Proyecto 2 Gabil Informe

Dos Ramos, Erick 07-40845
Formica, Gabriel 10-11036
Matteo Ferrando 09-10285
José Ponte 08-10893

Valle de Sartenejas

Resumen

El Proyecto consiste en aplicar un algoritmo genetico para la clasificacion entre 3 clases de flores Iris, las cuales son Iris Setosa, Iris Versicolour, Iris Virginica. Los datos para el aprendizaje cuentan con 4 características que son el ancho y largo tanto del pétalo como del sépalo.

Todo el algoritmo de aprendizaje y las funciones necesarias fueron implementadas por nosotros en el lenguaje Octave y no se utilizó ninguna librería para el desarrollo del proyecto.

Se implementaron 2 versiones de fitness, el tradicional basado en el porcentaje de clasificaciones correctas y otro en el que se penaliza el fitness del individuo por su tamaño.

Se implementaron 2 tipos de selección, el primero por rango en el que se eligen los mejores individuos según su fitness y el segundo es la selección por rueda de ruleta.

Las funciones de mutación y cruce también fueron implementadas, en el caso del cruce se realizó el de strings binarios de tamaño variable.

Descripción de la implementación

Todas las funciones involucradas en un algoritmo genético, en este caso Gabil, fueron implementadas por nosotros en el lenguaje Octave

Para la representación de los individuos se usó la representación de string binario, cada individuo tiene un tamaño múltiplo de 24, cada característica de un individuo presenta la siguiente distribución de bits.

Largo sepalo	Ancho sepalo	Largo Petalo	Ancho Petalo	Clase	Total
8	6	5	3	2	24

La distribución fue elegida de forma subjetiva observando el histograma de los datos por cada característica. El espaciado de cada característica es uniforme, es decir, se toma la resta entre el máximo valor y el mínimo valor de los datos por característica y se divide entre el número de bits asignado a ella.

Para la población inicial se generan individuos de tamaño máximo 72, es decir, cada individuo solo puede tener como máximo 3 conjuntos de reglas.

Se implementaron 2 versiones de fitness, el tradicional basado en el porcentaje de clasificaciones correctas y otro en el que se penaliza el fitness del individuo por su tamaño.

Se implementaron 2 tipos de selección, el primero por rango en el que se eligen los mejores individuos según su fitness y el segundo es la selección por rueda de ruleta.

Para la función de cruce se implementó el cruce para strings binarios de tamaño variable el cual es una generalización del cruce de 2 puntos.

La función de mutación toma cada individuo nuevo creado por la función de cruce y se modifica un bit de cada uno dado una probabilidad.

Descripción y análisis de los experimentos

Los experimentos se realizaron con diferentes parámetros para tasa de mutación y de cruce por cada función de fitness y selección. Las tasas de mutación tomando entre los valores [0.1, 0.5, 0.7] y la tasa de cruce entre los valores [0.3, 0.5, 0.7].

El conjunto de ejemplos utilizado para el aprendizaje es de 50 de los 150 datos ordenados de forma aleatoria y el resto de los datos se utilizan para validar los individuos.

Se tienen como límites un máximo de 100 generaciones o que el individuo con mayor fitness en la población de turno tenga un 98% de clasificaciones acertadas.

A continuación las tablas con los mejores resultados obtenidos para cada configuración.

Alg. Selección	Mejor Ind	Mejor Ind	Rueda de Ruleta	Rueda de Ruleta
Alg. Fitness	Normal	Penalizacion Tam	Normal	Penalizacion Tam
Tasa Mutación	0.7	0.7	0.7	0.5
Tasa Cruce	0.5	0.7	0.5	0.5
Tiempo (secs)	115	60	200	220
Generaciones	58	62	100	100
% Acertadas	100	82.23	26.52	20.26

Preguntas

a)Cuál es la mejor configuración de su algoritmo genético para clasificar los datos estudiados?

La mejor configuración obtenida con un máximo de 100 generaciones fue utilizando selección por mejor individuo, fitness estándar (porcentaje de clasificaciones correcta) con tasa de mutación de 0.7 y tasa de cruce de 0.5, obteniendo el 100% de clasificaciones correctas en solo 58 generaciones.

b)Cuál es el mejor conjunto de reglas hallado por el algoritmo genético? Considerando el número de ejemplos clasificados correcta e incorrectamente.

El individuo obtenido con la mejor configuración es el siguiente

Longitud: 360

Acertados (prueba): 100%

Acertados (validacion): 100%

c) Describa la función de fitness utilizada. Es útil incluir en la función de fitness un factor de penalización a clasificadores muy grandes?

La función de fitness utilizada es la estándar la cual retorna el porcentaje de clasificaciones correctas con respecto a un conjunto de ejemplos bien clasificados.

El fitness por factor de penalización en nuestras pruebas no dieron muy buenos resultados probablemente al aumentar el límite de generaciones posibles darían mejores resultados.