Algoritmos genéticos

Curso de Especialización en Inteligencia Artificial y Big Data

Modelos de Inteligencia Artificial

Introducción a la computación evolutiva

Los algoritmos evolutivos (AE) son **heurísticos** → Permiten encontrar **buenas soluciones en tiempos razonables** mediante el proceso de búsqueda de la solución final.

La heurística de los AE se inspira en la **evolución de los seres vivos** → Selección natural y genética.

Técnicas de **búsqueda** y **optimización**.

Negativo → Posible caída en óptimos locales.

Introducción a la computación evolutiva

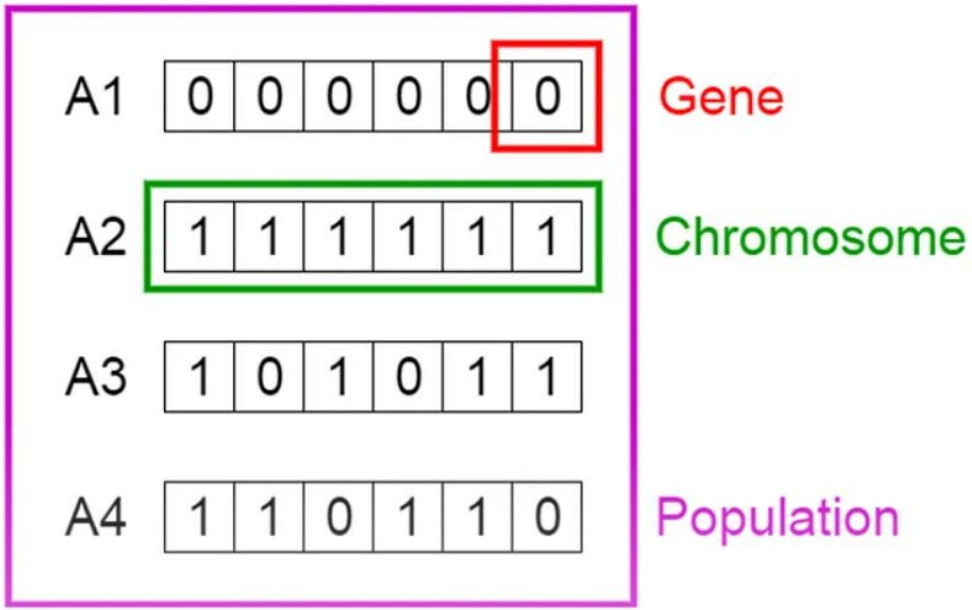
Un AE típico mantiene:

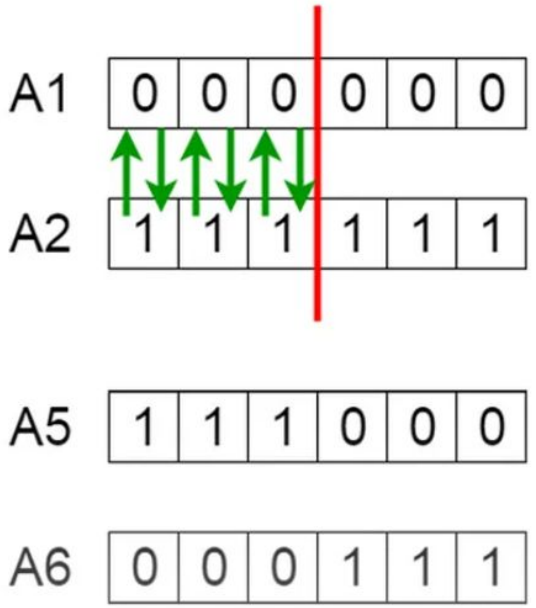
- **Población** → soluciones potenciales (individuos) de un problema. Evolucionan mediante reglas de selección, recombinación y mutación.

- **Fitness** → evaluación de cada individuo de la población. Nos da una medida de cuán bueno es el individuo dentro de la población.

- **Recombinación** y **mutación** → Modifican los individuos (heurística general de exploración)

Introducción a la computación evolutiva





https://towardsdatascience.com

Introducción a la computación evolutiva

Pasos básicos:

- Inicialización de la población, de forma aleatoria P(t)
- Proceso iterativo (transcurso de generaciones):
  - Selección de mejores individuos (mediante evaluación)
  - Recombinación y mutación → Da lugar a un nuevo conjunto de individuos
  - Evaluación de nuevos individuos
  - Generación de nueva población C(t)
  - Proceso de selección entre la nueva generación y la población

¿Qué es un algoritmo evolutivo?

Algoritmo 11.1 Algoritmo evolutivo.

1: $t \leftarrow 0$; /* generación inicial */
2: inicializar $P(t)$;
3: evaluar $P(t)$;
4: **mientras** (no se cumpla la condición de terminación) **hacer**
5: seleccionar padres de $P(t)$;
6: recombinar padres y mutar $\Rightarrow C(t)$;
7: evaluar $C(t)$;
8: seleccionar supervivientes de $P(t) \cup C(t) \Rightarrow P(t+1)$; /* sustitución generacional */
9: $t \leftarrow t+1$; /* siguiente generación */
10: **fin mientras**

Componentes de un algoritmo evolutivo

- Una **representación apropiada** de las soluciones del problema. Este punto es muy importante en el diseño de un AE ya que establece el espacio de búsqueda y de su elección depende la eficiencia del algoritmo.

- Una forma de crear una **población inicial de soluciones**. Aunque usualmente la población inicial se genera de forma aleatoria dentro del espacio de búsqueda, la incorporación de conocimiento puede ayudar a guiar la búsqueda.

Componentes de un algoritmo evolutivo

- Una **función de evaluación** capaz de medir la adecuación de cualquier solución, y que hará el papel de “entorno”, en el cual las mejores soluciones, esto es, aquellas con mejor adecuación, tengan mayor probabilidad de selección y supervivencia.

- Un **conjunto de operadores evolutivos** que actúan como reglas de transición probabilísticas (no deterministas) para guiar la búsqueda, y que combinan entre sí las soluciones existentes con el propósito de obtener otras nuevas.
-
- El valor de unos **parámetros de entrada** que el AE usa para guiar su evolución (tamaño de la población, número de iteraciones, probabilidades de aplicación de los operadores evolutivos, etc.).

Algoritmo genético simple

Representación:

- Cadenas binarias de bits (0’s y 1’s) → Cromosoma compuesto por genes
- Longitud determinada

$$c_i^t = (b_{i1}^t \dots b_{ilong}^t)$$

Algoritmo genético simple

Representación:

- Cadenas binarias de bits (0’s y 1’s) → Cromosoma compuesto por genes
- Longitud determinada

Un individuo de nuestra población se representa como:

$$X_i^t = (c_i^t, x_i^t, f_i^t)$$

Algoritmo genético simple

Obtención de la población inicial:

- Asignación de 0’s y 1’s a cada gen de los individuos
- Crear colección de m individuos

$$P(t) = \{X_i^t, \ldots, X_m^t\}$$

Algoritmo genético simple

Función de evaluación:

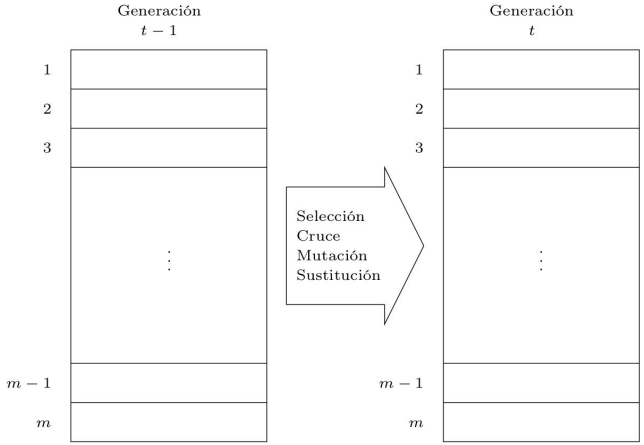
- Es la función objetivo del problema

$$f_i^t = eval(c_i^t) = f(x_i^t)$$

Algoritmo genético simple

Selección, muestreo, operadores genéticos y sustitución generacional:

1. Selección/muestreo de padres
2. Cruce
3. Mutación
4. Selección de supervivientes/sustitución generacional



Algoritmo genético simple

Esquemas de selección:

- Por **ranking**:
  - Ordenamos los individuos en una lista y la probabilidad de selección de un individuo se obtiene según su posición en la lista ordenada.

$$p_i^t = (a_{max} - (a_{max} - a_{min}) \cdot \frac{rank(c_i^t) - 1}{m - 1}) \cdot \frac{1}{m}$$

rank es la posición en la lista ordenada

amin = amax − 1

1 <= amax <= amin

$$p_i^t = q \cdot (1 - q)^{rank(c_i^t) - 1}$$

Otra opción: q es un valor introducido por el usuario

Algoritmo genético simple

- **Por torneo**

Se selecciona el individuo con mejor adecuación de un grupo de individuos (2 en el torneo binario) elegidos aleatoriamente de la población.

Algoritmo genético simple

Operaciones de cruce:

- **Cruce uniforme** → máscara binaria de cruce
  - Si 1 → Tomamos el valor del primer padre
  - Si 0 → Tomamos el valor del segundo padre
- **Cruce aritmético** → genes son variables continuas
  - Cada gen de un cromosoma se calcula como:
    - ci = a * bi + (1 - a) * bj
- **Cruce plano** → cruce por un punto
  - Creamos un punto de pivote k gracias al cual vamos a utilizar los valores de la posición 1 hasta k del primer individuo y el resto del segundo para generar un primer hijo, el otro hijo se genera utilizando la inversa de lo generado para el primero.

Algoritmo genético simple

Operaciones de mutación:

- **Mutación uniforme** → Cambio del gen a mutar por un valor generado de forma aleatoria en el dominio de la variable.
- **Mutación por intercambio** → Intercambio de dos elementos elegidos aleatoriamente de una permutación.