
Computación Bioinspirada - Práctica N° 10

PROFESOR DEL CURSO: Dennis Barrios Aranibar

FECHA: 5 de Noviembre del 2018

ASISTENTE DEL CURSO: Kevin Christian Rodríguez Siu

Objetivos de la Sesión

- Conocer una de las técnicas basadas en Sistemas Inmunológicos Artificiales
- Utilizar el Algoritmo de Selección Negativa para poder detectar elementos discordantes con un conjunto.

Contexto

En las figuras se ilustra el proceso del Algoritmo de Selección Negativa. Se basa en la generación de patrones o detectores de errores y después, en la comprobación de un conjunto de cadenas protegidas, con el fin de revisar si han tenido o no algún cambio.

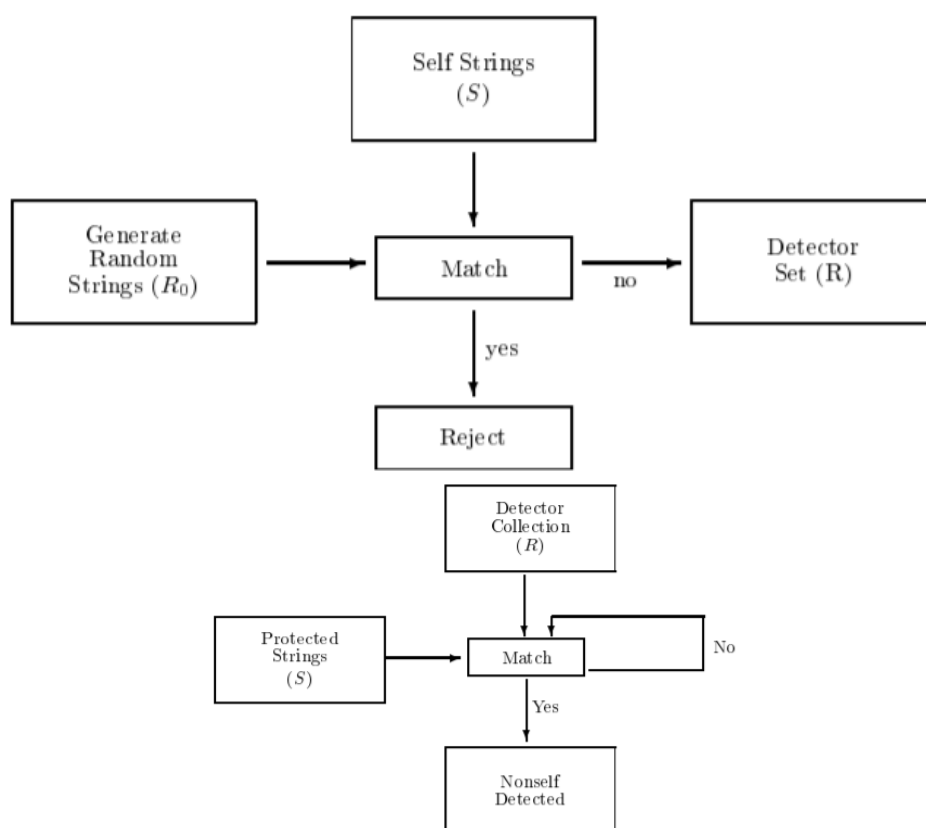


Figura 1: Esquema del Algoritmo de Selección Negativa: Generación y Aplicación de Detectores

Ejercicios

Realizar una implementación del Algoritmo de Selección Negativa para poder hacer la detección de colores con cierta cantidad de tonalidad azul, que vamos a definir por la siguiente fórmula:

$$tonoAzul(r, g, b) = \begin{cases} \text{si } r \leq g & b \geq 128 \wedge abs(r - g) < 85 \wedge \frac{r+g}{2} \leq 128 \\ \text{si } r > g & b \geq 128 \wedge abs(r - g) < 85 \wedge \frac{r+g}{2} \leq 85 \end{cases} \quad (1)$$

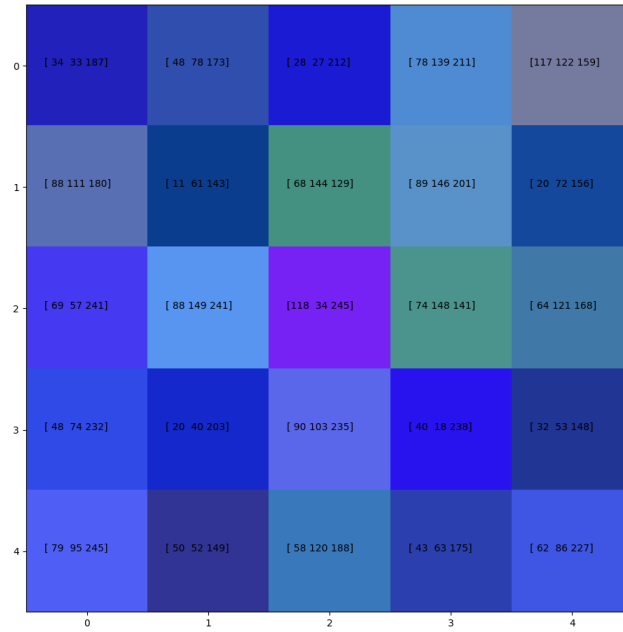


Figura 2: Ejemplo de Colores aceptados como azul por la fórmula

1. Debe definir todos los elementos de la técnica a programar. Esto incluye:
 - **Self-Strings**, que es la definición de un conjunto de datos aceptado. Estas deben ser representaciones binarias (24 bits) de colores que cumplen la fórmula *tonoAzul*.
 - **Función de Generación de Detectores**, que van a surgir del emparejamiento de cadenas aleatorias que no coinciden con ninguna de las cadenas definidas en el conjunto de **Self-Strings**.
2. Una vez hecho esto, realiza el proceso del algoritmo:
 - (a) Generación del conjunto de datos aceptado, de tamaño $n \geq 100$.
 - (b) Generación de m detectores. Recuerda que sólo se aceptan los patrones de bits que no coinciden con ningún otro.
 - (c) Aplicar los detectores con un conjunto de colores generados aleatoriamente. Si hay alguna coincidencia, el color es considerado como azul. Sino, se considera como una variación / mutación, por tanto no se considera como azul.
3. Por cada color aleatorio detectado, se debe calcular el resultado del Algoritmo de Selección Negativa (utilizando la cadena de bits del color) y el resultado de la formula *tonoAzul*. Si ambas coinciden, se cuenta como una predicción exitosa.
4. Al final, el programa debe devolver el porcentaje de predicciones exitosas en relación al total de predicciones.

No es necesaria una interfaz gráfica avanzada, pero debe existir una visualización de lo que ocurre, que debe incluir el valor del color que se prueba (en formato RGB), el resultado de la aplicación de los detectores y si este se corresponde con la fórmula definida por la función

$\text{tonoAzul}(r, g, b)$, indicando si los resultados de ambas pruebas coinciden o no.

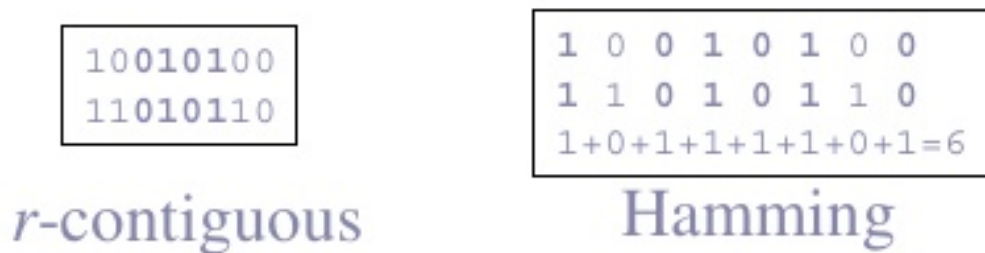


Figura 3: Ejemplo de Distancia de Hamming y de Emparejamiento por r -bits

Actividades

1. Genera 50 colores aleatorios. Realiza la generación de 10, 30, 50 y 100 detectores utilizando una distancia de Hamming (haciendo un emparejamiento de bits, y contando cuantos bits son iguales entre dos cadenas) con valores 13,15 y 17. Cada combinación debe utilizar la misma lista de colores aleatorios ¿Con qué combinación de cantidad de detectores y valor de distancia de Hamming obtienes un mayor porcentaje de predicciones exitosas?
2. Genera 50 colores aleatorios. Realiza la generación de 10, 30, 50 y 100 detectores utilizando ahora un emparejamiento de $r - bits$ contiguos, con un valor r de 13, 15 y 17. Cada combinación debe utilizar la misma lista de colores aleatorios ¿Con qué combinación de cantidad de detectores y valor de $r - bits$ contiguos obtienes un mayor porcentaje de predicciones exitosas?
3. ¿Con qué técnica obtuviste mejores resultados: Distancia de Hamming o $r - bits$ contiguos? Explica brevemente por qué crees que estos resultados se dieron así.

```
44 (53, 9, 171) -> Por Seleccion Negativa: False . Por formula: True
45 (111, 49, 198) -> Por Seleccion Negativa: True . Por formula: True
46 (32, 69, 128) -> Por Seleccion Negativa: True . Por formula: True
47 (64, 63, 218) -> Por Seleccion Negativa: True . Por formula: True
48 (45, 44, 95) -> Por Seleccion Negativa: True . Por formula: False
49 (121, 180, 106) -> Por Seleccion Negativa: False . Por formula: False
50 (229, 171, 40) -> Por Seleccion Negativa: False . Por formula: False
51 Porcentaje de Exito: 90.0
```

Figura 4: Ejemplo del Formato de Salida: Indica los colores, los resultados tanto del algoritmo como de la fórmula, y el porcentaje de éxito de al final.

Desarrollo y Entrega

- El trabajo debe ser desarrollado en la sesión de laboratorio.
- Se debe entregar digitalmente (en un PDF vía email de preferencia) un informe conteniendo el desarrollo de todas las actividades y los códigos implementados.
- Plazo de entrega del informe: 5 de Noviembre del 2018.

Cuadro 1: Rúbrica Práctica N° 10

Criterio	Modelamiento del Problema	Selección y Fundamentación de Elementos de la Técnica	Ejecución de la Técnica y Código Fuente	Resultados y Visualización (Actividad 1 y 2)	Informe (Actividad 3)
Nulo (0%)	No existe el modelado del problema	No existe la selección de los elementos de la técnica.	No hay código fuente que muestre la ejecución.	No hay resultados visibles.	No se realizó informe de estas actividades.
Deficiente (25%)	Se han definido algunos aspectos del problema a resolver de forma difusa.	Se han seleccionado algunos elementos de la técnica solicitada al azar. No están implementados en el código fuente.	El código existe, pero no es ejecutable.	Hay una muestra del proceso de ejecución pero no de los resultados.	Se ha realizado un 25% o menos de las preguntas adicionales. No hay un formato u orden en el informe.
Regular (50%)	Se han definido los aspectos del problema a resolver claramente.	Se han seleccionado todos los elementos de la técnica, pero no todos han sido justificados. No todos han sido implementados correctamente en el código.	Existe código fuente ejecutable, que tiene algunas nociones de los requerimientos del problema.	Hay muestra del proceso de ejecución y de los resultados, pero estos no son fáciles de entender.	Se ha realizado un 50% o menos de las preguntas adicionales. El informe tiene cierto orden en algunas secciones.
Bueno (75%)	Se han definido los aspectos del problema a resolver de forma clara y se identifica su función dentro del proceso de la técnica.	Se han seleccionado todos los elementos de la técnica, y todos tienen algún tipo de justificación. Se han implementado de forma regular en el código fuente.	Existe código fuente ejecutable que cubre los requerimientos del problema, ejecuta la técnica pedida y que muestra algún tipo resultados.	Hay muestra del proceso de ejecución y de los resultados según el formato solicitado.	Se han realizado todas las preguntas adicionales correctamente. El informe se divide claramente en secciones.
Excelente (100%)	Se han definido los aspectos del problema a resolver de forma clara y se identifica su función dentro del proceso de la técnica, haciendo una implementación de acuerdo a lo solicitado en los ejercicios.	Se han seleccionado todos los elementos de la técnica solicitada y todos tienen una justificación clara, que muestra claramente la idea que tiene por detrás y su implementación en el código fuente.	Existe código fuente ejecutable y fácilmente legible que cubre los requerimientos del problema, ejecuta la técnica pedida y muestra resultados de acuerdo a lo solicitado en la práctica.	Hay muestra del proceso de ejecución y de los resultados según el formato solicitado, existiendo además una breve discusión sobre los mismos.	Se han respondido las preguntas adicionales con muy buenas respuestas, claras y fundamentadas. El informe cumple con las disposiciones de formato y orden de la Universidad.
Puntaje Máximo	2.5	5.5	5	3.5	3.5