Tarea 1 de Computación 1 (2021)

Ingeniería Forestal, CENUR Noreste

Descripción del problema

En teoría de la información, la distancia de Hamming[1] es la distancia entre dos palabras válidas de un código (o un conjunto predefinido de palabras válidas). La distancia entre dos palabras p1 y p2 en código binario (compuesta por ceros y unos) equivale a la cantidad de posiciones de p1 (bits) que deben cambiarse para transformar a p1 en p2. Cuanta más distancia exista entre las palabras de un código, es menos probable que una palabra válida, al intercambiar ceros por unos debidos a errores de transmisión, se transforme en otra palabra válida, por lo que la distancia de Hamming es utilizada para detectar y corregir estos errores.

Ejemplos

- La distancia Hamming entre 1011101 y 1001001 es 2.
- La distancia Hamming entre 2143896 y 2233796 es 3.
- La distancia Hamming entre "tener" y "reses" es 3.

Aplicada a vectores de números enteros, la distancia entre dos vectores es la cantidad de posiciones que difieren. Por ejemplo la distancia de Hamming entre p1=[1,2,3,4,3,2,3,4] y p2=[9,2,3,4,0,2,3,1] es 3, ya que hay tres posiciones que difieren (la 1, la 5 y la 8).

Se pide desarrollar en Octave las siguientes funciones:

• **distHamming**, que recibe dos vectores de dígitos y devuelve la distancia de Hamming entre ambos vectores. Si los vectores son de distinto largo o alguno de los vectores es vacío, la función debe devolver -1.

• **encontrarMasParecido**, que recibe dos vectores de dígitos v1 y v2 y retorna:
  – (1er parámetro de salida) la posición de comienzo en v1 de la secuencia, del largo de v2, con menor distancia de Hamming a v2.
  – (2do parámetro de salida) la distancia de Hamming entre v2 y la secuencia encontrada.
  – Si v2 es más largo que v1 o alguno de los vectores es vacío, la función debe devolver -1 en ambos parámetros.
  – Si hay más de una secuencia con distancia mínima a v2 debe devolverse la posición en v1 más pequeña.

Ejemplos:

• dist = distHamming([2,3,4,2],[2,3,4,2]) devuelve dist=0

• dist = distHamming([2,3,4,2],[1,3,9,2]) devuelve dist=2

• [pos,dist] = encontrarMasParecido([1,2,3,4,3,4,2,5,6,7,3,2],[3,4]) devuelve pos=3 y dist=0

• [pos,dist] = encontrarMasParecido([1,2,3,4,3,4,2,5,6,7,3,2],[3,4,9,4]) devuelve pos=3 y dist=1

1 Distancia de Hamming: https://es.wikipedia.org/wiki/Distancia_de_Hamming

Validación

• Los valores válidos de entrada son vectores de cualquier largo cuyas entradas son dígitos entre 0 y 9. No se evaluará el comportamiento de la entrega ante entradas inválidas.
• Es imprescindible para el proceso de validación que las funciones respeten los nombres, la cantidad y el orden de los parámetros de entrada y salida establecidos en la letra.

Se pide

Entregar 2 archivos .m separados cada uno con los siguientes nombres:

- **distHamming.m** (contiene la función distHamming)
- **encontrarMasParecido.m** (contiene la función encontrarMasParecido)

Importante

- No se debe entregar un archivo comprimido sino que los 2 archivos separados.
- Para la corrección, las tareas se ejecutarán con la versión 6+ de Octave.
- La ejecución se realizará desde la línea de comandos (sin interfaz gráfica).
- No está permitido utilizar facilidades de Octave que permitan resolver los problemas de forma trivial (funciones max, min, sort, etc.)
- En esta tarea, como en todos los problemas de este curso, se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. De esta manera, se hará énfasis en buenas prácticas de programación que lleven a un código legible y bien documentado, tales como:
    - indentación adecuada
    - utilización correcta y apropiada de las estructuras de control
    - código claro y legible
    - algoritmos razonablemente eficientes
    - utilización de comentarios que documenten y complementen el código
    - nombres mnemotécnicos para variables, constantes, etc.