

Comunicaciones Digitales

Taller 3

Teoría de Información: Codificación

Instituto de Ingeniería Eléctrica

Parte I: Canal simétrico con borrado

En este ejercicio trataremos de evaluar qué ventajas puede tener enriquecer el bloque de decisión con la posibilidad de contestar “no sé” ante una entrada muy sobre el borde de la región de decisión. Esto resulta en lo que se denomina habitualmente canal con borrado, si bien el término resulta un tanto engañoso pues el decodificador tomará en cuenta este nuevo símbolo.

Formalmente, un canal binario simétrico con borrado se define de la siguiente manera. La fuente X emite símbolos de un alfabeto $A = \{x_0, x_1\}$. En el canal se introduce ruido, y el receptor decide entre 3 símbolos posibles a la salida: la variable aleatoria Y en el receptor toma valores en el alfabeto $B = \{x_0, e, x_1\}$. El símbolo e (del inglés *erasure*) corresponde a una zona de detección entre el x_0 y el x_1 donde el receptor no se arriesga por x_0 o x_1 .

Los símbolos a la entrada del canal se generan en forma independiente y con probabilidades:

$$\begin{aligned} P(X = x_0) &= a \\ P(X = x_1) &= 1 - a \end{aligned} \quad (1)$$

Las probabilidades condicionales con que se reciben los símbolos son:

$$\begin{aligned} P(Y = x_0|X = x_0) &= P(Y = x_1|X = x_1) = 1 - p - q \\ P(Y = x_0|X = x_1) &= P(Y = x_1|X = x_0) = q \\ P(Y = e|X = x_0) &= P(Y = e|X = x_1) = p \end{aligned} \quad (2)$$

- Hallar la probabilidad de que un símbolo sea *borrado* por el algoritmo de detección.
- Calcular los bits de información por símbolo transmitido. Graficar en función de a .
- ¿En qué condiciones se obtiene máxima incertidumbre sobre el símbolo que se va a transmitir? Indicar este resultado en la gráfica anterior.
- ¿Se podría modificar un sistema BPSK de tal manera que este modelo de canal sea adecuado? Dada su modificación, y considerando un canal AWGN, ¿cómo calcularía las probabilidades p y q ?




Nos enfocaremos en una situación de SNR razonable, por lo que p será pequeño.

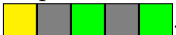
- Dado lo anterior, ¿le parece razonable asumir que q es despreciable? Justifique y en lo que sigue actúe en consecuencia.

- (f) Calcular las funciones $\mathbb{H}\{Y\}$, $\mathbb{H}\{Y|X\}$, $\mathbb{H}\{X|Y\}$, e $I\{X;Y\}$. Dar una interpretación intuitiva a estas expresiones. ¿Cuánto vale la capacidad de este canal?
- (g) Compare la capacidad del canal anterior con el del canal BPSK “tradicional” (es decir, con solo dos símbolos a la salida) para idénticas condiciones de ruido y potencia de señal. Es decir, para ciertos valores de tasa de error de bits en el BPSK original, calcule p (y q) para el sistema modificado (para lo que deberá usar su diseño de la parte (d)) y compare las capacidades resultantes. ¿Es más conveniente este sistema modificado?

Parte II: Algoritmos de decisión iterativa¹

El wordle es un juego web (<https://www.nytimes.com/games/wordle/index.html>) donde el objetivo es adivinar en un máximo de seis intentos una palabra en inglés de cinco letras. En cada intento, el juego nos va a devolver un patrón de colores, indicando para cada letra si:

-  La letra está en el lugar correcto.
-  La letra está en la palabra, pero en el lugar incorrecto.
-  La letra no está en la palabra.

Por ejemplo, si la palabra secreta es **scale** y nosotros ingresamos **crane** el juego nos devolverá . Esto pues la **c** está en la palabra pero no en el primer lugar, la **r** y la **n** no están en **scale**, y tanto la **a** como la **e** están en el lugar correcto. Es importante remarcar que no se puede ingresar cualquier palabra (deben existir en una lista bastante extensa) y la solución (que cambia todos los días) siempre pertenece a esta lista.

Un ser humano juega básicamente abriendo con una palabra, y para el segundo intento busca en su vocabulario una palabra que cierre con los patrones que vio hasta el momento. Siguiendo con el ejemplo anterior, la siguiente palabra podría ser **place**, que cierra con lo que sabemos hasta el momento (deja quietos la **a** y la **e**, cambia de lugar la **c** y prueba con la **p** y la **l**).

Como estudiantes de ingeniería que somos, vamos a pensar matemáticamente en el problema y cuál sería una buena forma de resolverlo. En particular, atacaremos el problema usando teoría de información. Sea X el símbolo aleatorio que representa la palabra oculta. Sean x_k la palabra que ingresamos en el intento k -ésimo, Y_k el patrón (aleatorio) correspondiente y \mathcal{X}_k el conjunto de palabras que concuerdan con los patrones y_1, y_2, \dots, y_k que hemos observado hasta ahora.

- (a) Argumente por qué Y_k , si bien es aleatorio, es una función de X y x_k . De ahora en más, para hacer esto explícito puede usar $Y_k = f(X, x_k)$
- (b) ¿Qué representa intuitivamente $\mathbb{H}\{X|Y_k\}$?

Trataremos de darle sentido cuantitativo a lo anterior.

¹Este ejercicio está basado en el video <https://youtu.be/v68zYyaEmEA> del usuario 3Blue1Brown y el código que comparte en https://github.com/3b1b/videos/tree/master/_2022/wordle

- (c) Halle una expresión que relacione $\mathbb{H}\{X|Y_k = y_k\}$ con el tamaño del \mathcal{X}_k resultante de observar $Y_k = y_k$. Puede suponer que, al momento de hacer el intento k -ésimo, la palabra oculta X se encuentra uniformemente distribuida en \mathcal{X}_{k-1} . En base a lo anterior, escriba $\mathbb{H}\{X|Y_k\}$ en términos de una esperanza respecto del tamaño de \mathcal{X}_k .
- (d) Dada su respuesta a las dos últimas partes, argumente por qué nuestra elección de x_k debería buscar maximizar la entropía de Y_k . Para eso, desarrolle la entropía condicional y aplique la observación de la parte (a).

Usaremos una implementación de lo anterior para verificar los beneficios de este algoritmo sencillo.

- (e) Siendo \mathcal{X}_k las palabras que concuerdan con los patrones vistos hasta ahora, ¿cómo podría calcular la probabilidad de cada patrón para cierto $x_k \in \mathcal{X}_k$? ¿Cómo podría entonces calcular la entropía de $Y_k = f(X, x_k)$ para todos los $x_k \in \mathcal{X}_k$?

En el EVA del curso se puede encontrar un programa en Python con el que probar este algoritmo (i.e. elige $\operatorname{argmax}_{x_k \in \mathcal{X}_k} \mathbb{H}\{f(X, x_k)\}$). También cuenta con una implementación que elige al azar entre cualquiera de las palabras en \mathcal{X}_k , simulando así un ser humano que busca una palabra cualquiera que cierre con los patrones. La única diferencia con el juego es que permite una cantidad ilimitada de intentos. El programa evalúa el desempeño del algoritmo usando como solución todas las posibles palabras una por una, y reporta la cantidad promedio de intentos que necesitó el método para hallar la respuesta. Cuidado porque la primer palabra debe elegirla usted, e influirá naturalmente el desempeño del algoritmo.

- (f) Corra ambas versiones (al azar y maximizando la entropía) y reporte el desempeño de cada método. Verifique que el puntaje obtenido por el aleatorio es similar al de un humano.

Parte III: Compresión de la fuente

En esta parte se trabajará con extensiones de la fuente y el algoritmo de Huffman para verificar su capacidad de compresión. Se usarán los archivos de texto `adams.txt`, `mensaje1.txt` y `mensaje2.txt` como fuentes. Se utilizará un script de Python a descargar del EVA del curso para realizar la compresión.

Para el archivo indicado en la variable `archivotxt` el script primero lo interpreta como binario, luego toma extensiones de ese alfabeto de distintos órdenes, calcula la distribución empírica de los símbolos en el mensaje, calcula la entropía resultante y finalmente aplica el algoritmo de Huffman para hallar el código “óptimo”. Además, calcula la tasa de compresión del código comparando el tamaño en bits del archivo codificado y el original.

- (a) ¿Porqué la palabra *óptimo* está entre comillas en el párrafo anterior? ¿Qué hipótesis del algoritmo de Huffman no se cumple?
- (b) Observe las entropías resultantes para los tres archivos usando las diferentes extensiones de la fuente, y compárelas entre ellas. Modifique el código de manera de graficar la distribución empírica, y use alguna de estas gráficas para explicar los resultados obtenidos.

- (c) Modifique el código para que para cada extensión de fuente, imprima el código de Huffman por orden del tamaño de palabra codificada. ¿Qué observa para los distintos archivos?
- (d) Compare el tamaño del mensaje codificado para los diferentes archivos y explique los resultados obtenidos. A modo de referencia, tome el archivo original y comprímalo usando alguna herramienta de su sistema operativo (como generar un `zip` o similar). ¿El algoritmo de Huffman logra resultados similares? ¿Es correcto tomar en cuenta únicamente el tamaño del archivo codificado por Huffman al momento de comparar?

Parte IV: Codificación para corrección de errores

- (a) Realizar al menos los ejercicios 9.4, 9.6 y 9.7 del capítulo 9 de las notas de Belzarena y Larroca. En particular, en el ejercicio 9.7 se necesita un archivo en formato `wav`. Hay una variada selección en <https://iie.fing.edu.uy/ense/asign/tallerinte/repo/canciones/>. Además, la versión con código corrector de errores que pide la parte (c) se la brindamos ya implementada para su comodidad.

Referencias

Principles of Digital Communications, Robert Gallager, 2008.

Comunicaciones Inalámbricas, Pablo Belzarena y Federico Larroca, 2017.

Consultas

En las clases de consulta de *Comunicaciones Digitales* (online) o en el foro del curso.