

Trabajo Práctico Especial

Temperaturas, Grump y Transmisiones: El Mundo en Entropía

Introducción

En un mundo afectado por crisis climáticas, tarifas comerciales y tweets espaciales de Elon Musk, la información es poder... o al menos, datos con estilo. Este trabajo práctico te invita a explorar el fascinante universo de la Teoría de la Información, analizando algo tan simple —y, paradójicamente, tan geopolíticamente condicionado por mapas y climas— como las **temperaturas promedio** de tres ciudades con condiciones meteorológicas (y políticas) bien distintas.

Contexto Geopolítico y Científico

Supongamos que Elon Musk decide lanzar una nueva constelación de satélites para monitorear las temperaturas del mundo en tiempo real y vender los datos como NFT. El sistema tiene tres sensores ubicados estratégicamente en:

- **Quito:** Donde la temperatura no cambia ni aunque recen diez climas distintos.
- **Melbourne:** Donde podés experimentar las cuatro estaciones antes del almuerzo.
- **Oslo:** Donde el clima no se decide si quiere ser Siberia o un spa nórdico.

Pero con las tarifas impuestas por un tal presidente Grump por cada bit transmitido desde el espacio, las transmisiones están comprometidas. Como ingeniero/a del sistema, tu misión es analizar, codificar, comprimir y ayudar a predecir los datos... sin que Grump te deje en la bancarrota de bits.

Para esto, deberás resolver los siguientes ítems en forma computacional y analizar los resultados obtenidos, de acuerdo a las pautas que se indican:

Parte 1: Estadísticas para ingenieros que miran el cielo

Dadas las señales de temperaturas diarias registradas durante cierto periodo en las tres ciudades (S_1 : [Quito](#), S_2 : [Melbourne](#), S_3 : [Oslo](#)), expresadas como valores enteros, en °C (grados centígrados):

- 1.1 Calcular la temperatura **promedio** y la **desviación estándar** para cada señal S_i y analizar cómo se comportan estadísticamente.
- 1.2 Calcular el **factor de correlación cruzada** entre cada par de señales. Discutir si existen correlaciones significativas o no (*tratando de establecer, por ejemplo, si **Melbourne** podría estar prediciendo el clima de **Quito**, o de **Oslo**.. o si no tienen nada que ver*).

Parte 2: Una fuente de calor... markoviana

Considerando los valores de temperatura t que componen cada señal S_i , construir una nueva señal T_i compuesta por una secuencia de símbolos discretos **F**, **T** o **C**, definidos según:

- **F** (frío): si $t < 11^{\circ}\text{C}$
- **T** (templado): si $11 \leq t < 19^{\circ}\text{C}$
- **C** (cálido): si $t \geq 19^{\circ}\text{C}$

Para cada T_i :

1. Modelar la fuentes con memoria de orden 1 (Markov), obtener la matriz de transición y analizar su comportamiento (*por ejemplo, tratá de descubrir cosas como: En Oslo, si hace frío hoy, es casi seguro que siga así hasta julio..*)
2. Usar **muestreo Monte Carlo** para obtener, para cada símbolo:
 - La **probabilidad estacionaria** (esa a la que llegás después de mucho simular).
 - El **tiempo medio de 1° recurrencia** (ese que te dice, en promedio, cuánto tarda un símbolo en volver a aparecer después de haberse emitido).

Nota: Experimentar con diferentes umbrales de convergencia ϵ (*comentar si realmente influyen en los resultados, o todo es una ilusión matemática*). Analizar precisión de resultados en función del tiempo e incluir gráfico de convergencia.

Parte 3: Entropía, Huffman y la batalla por los bits

1. Calcular la **entropía** de cada fuente T_i :
 - sin memoria (orden 0), considerando símbolos individuales.
 - con memoria (orden 1), usando información sobre transiciones entre símbolos.

Interpretar los resultados: ¿Qué ciudad presenta menor entropía? ¿Cuál más? ¿Esto hace que alguna ciudad sea más impredecible que otra? (*Spoiler: tal vez no sea la que imaginas*).

2. Implementar el algoritmo de **Huffman** para codificar cada señal T_i y su extensión a orden 2, teniendo en cuenta que la fuente es markoviana. Aplicar el Teorema de **Shannon** y analizar resultados (*Shannon se revuelca en su tumba? o aplaude desde el más allá?*).
3. En cada caso, calcular la longitud total del mensaje codificado (en bits), compararla con la longitud original del archivo y obtener la tan ansiada **tasa de compresión**.

Nota: no hace falta que guardes el archivo comprimido (*pero actuá como si te importara*).

Parte 4: El canal climático de Musk

Ahora el satélite en órbita (SpaceHeat-42) transmite la señal S_2 (de **Melbourne**)... pero lo que llega a la base terrestre es S_4 (Melbourne "ruidoso"), misteriosamente diferente.

1. Generar T_4 (de igual manera igual que se generaron las otras T_i), y construir la matriz de canal comparando T_2 (entrada) y T_4 (salida).
2. Calcula el **Ruido** del canal y la **Información mutua**. Analizar los valores obtenidos (*explicar si es un buen canal o una porquería disfrazada de innovación*).

Pautas de desarrollo y entrega

(aprobada por el comité académico y el algoritmo de YouTube)

Importante: Recordá que para promocionar la materia, es requisito aprobar este Trabajo Práctico Especial (TPE) en formato grupal, y su correspondiente defensa individual, con un mínimo de 7 (Siete. *No seis con carita triste... siete*). Además, también necesitás sacar 7 o más en el parcial. (*Si no llegás a esos 7 gloriosos puntos, tranqui: no perdés la cursada – salvo que desapruebes todas las instancias del parcial–, pero no podrás promocionar y tendrás que rendir final*).

Modalidad de entrega:

- El TPE se realizará en grupos de 2 o 3 personas (*Más personas, más conflictos internos. Menos, menor chance de discutir ideas y enfoques, o encontrar soluciones juntos*).
- La entrega deberá realizarse en Moodle, por parte de uno de los integrantes de cada grupo (indicando claramente apellido y nombre de quienes lo componen).
- Fecha límite: **19/06/25** a las 23:59 hs. (*sí, exactamente a las 23:59, no a las 00:00 porque eso ya es el día siguiente... y tu grupo pasará a ser leyenda urbana: “el que casi entrega”*).

¿Qué tenés que subir a Moodle? Un **.zip** (menos de 10MB, porque no vamos a almacenar un datacenter de SpaceX), que contenga:

1. Un informe en PDF (máx. 10 páginas, *formato humanoide, sin márgenes de 7 cm ni letra tamaño cartel de autopista*):
 - Carátula, incluyendo apellidos y nombres de integrantes y email de contacto.
 - Introducción: descripción breve del problema. (*Si no sabés qué poner, releé el enunciado hasta que tenga sentido*).
 - Desarrollo y análisis: explicación clara (y *honest*) de qué hicieron, cómo lo hicieron, por qué lo hicieron así. Incluyan pseudocódigos, gráficos, análisis de resultados y reflexiones (*no es necesario ponerse filosóficos*).
 - Conclusiones: resumen de lo aprendido, lo que funcionó, lo que no (y *si sobrevivieron a Montecarlo sin perder la fe*).
2. Código fuente + ejecutable (*Por favor, no nos obliguen a adivinar qué librerías faltan ni a instalar software de dudosa procedencia*). Verificá que no uses rutas absolutas como `C:\MisTrabajos\TemperaturaFinal_FINAL_revisado_FINAL_v3\`.
3. Link al código ejecutable en alguna plataforma online como [replit](#) o Google Colab.
Nota: *No toquen nada después de la fecha de entrega. No hagan “hotfixes” ninja. Sabemos mirar el historial.*

Defensa del TPE:

La defensa (para quienes aspiren a promocionar) será individual y se realizará de manera presencial el **03/07/25** (*No basta con que alguno del grupo haya hecho todo*).

Algunas recomendaciones adicionales:

- Los resultados que incluyas deben estar justificados (*Si aparecen números mágicamente sin explicación, los vamos a cuestionar como si estuviéramos investigando una licitación pública*).

- No incluir el código fuente ni cálculos o tablas auxiliares en el informe (*Si creés que necesitás agregarlo, ponelo como apéndice...pero cuenta dentro del límite de 10 páginas!*).

Nota Final: Este TPE no solo mide tu conocimiento técnico, sino también tu capacidad para colaborar en grupo, discutir y comunicar ideas, sobrevivir a algoritmos estocásticos sin entrar en crisis existencial y entregar el trabajo sin dramas dignos de una serie de Netflix.