

Memoria

Blanca Cano Camarero y Iker Villegas Labairu.

4 de noviembre de 2022

Indice de contenidos

Prefacio	4
1 Introducción	5
2 Cálculo de <i>spikes</i>	6
2.1 Lectura de los datos	6
2.1.1 Descripción	6
2.1.2 Requisitos	6
2.2 Lectura de los datos	7
2.3 Diseño del algoritmo de cálculo de <i>spikes</i>	7
2.3.1 Motivación del algoritmo	8
2.4 Determinación de los umbrales	9
3 Cálculo de la información mutua	16
3.1 Abstracción del problema	16
3.2 Cálculo de la información mutua	16
3.3 Formulación del experimento	17
4 Cálculo de entropía normaliza o información mutua normalizada	20
4.1 Resultados obtenidos	20
4.1.1 Trozo C	20
4.1.2 Trozo R	22
4.1.3 Trozo G	23
5 Suposición de otro tipo de codificación de eventos	24
5.1 Descripción del sistema de codificación SAX	24
5.2 Observaciones	25
5.3 Sobre nuestra implementación	25
5.4 Experimentación con SAX	25
5.4.1 Descripción del experimento	25
5.4.2 Resultados obtenidos	26
5.4.3 Trozo C	26
5.5 Comparativa	31
5.5.1 Ventajas	31
5.5.2 Desventajas	31
5.6 Trozo R	31

5.7	Trozo G	33
5.8	Conclusiones trozo C y G	33
5.9	Futuros campos que contemplar esta codificación	33
5.10	Notas	35
6	Summary	36
	References	37
7	Apéndice	38
7.1	Resultado experimentos para la determinación de los umbrales	38
7.1.1	Calculando umbrales de ./DatosSinapsisArtificial/TrozoC.zip para la neurona LP	38
7.1.2	Calculando umbrales de ./DatosSinapsisArtificial/TrozoC.zip para la neurona VD	40
7.1.3	Calculando umbrales de ./DatosSinapsisArtificial/TrozoR.zip para la neurona LP	41
7.1.4	Calculando umbrales de ./DatosSinapsisArtificial/TrozoR.zip para la neurona VD	43
7.1.5	Calculando umbrales de ./DatosSinapsisArtificial/TrozoG.zip para la neurona LP	45
7.1.6	Calculando umbrales de ./DatosSinapsisArtificial/TrozoG.zip para la neurona VD	47
7.2	Información mutua obtenida para la codificación SAX	48
7.2.1	Trozo A	48
7.2.2	Trozo R	52
7.2.3	Trozo G	56

Prefacio

Práctica de la asignatura de Teoría de la Información del máster de Ciencia de Datos de la UAM del curso 2022-2023.

Esta memoria ha sido generada con [Quarto](#) y el lenguaje utilizado ha sido [Python](#).

La estructura de directorio es la siguiente:

- **src** contiene los ejecutables. Cada fichero se encarga de una función concreta, ya sea una colección de funciones o la implementación de un experimento. A lo largo de la memoria se detallará su contenido y se explicará cómo llamarlos.
- **experiment_results** Contiene los resultados de los experimentos de ejecutables del **src**.
- **img** Contiene las imágenes generadas durante los experimentos. En la memoria se muestran las más interesantes, pero no todas.

Los notebooks en formato **.qmd** es la memoria ejecutable, se comporta como un jupyter.

Para poder ejecutar la memoria entera dispone de un **Makefile** cuyas funciones básicas son:

- **make** o **make render** Para renderizar un pdf.
- Visualización de la memoria en html **make preview**.

1 Introducción

Cálculo de Información Mutua entre dos series temporales

A lo largo de esta memoria se calculará la Información Mutua (IM) entre las series temporales de la neurona VD y la neurona LP, en estados de control, inyección de Gaba y recuperación del sistema. Así como se calculará la transferencia de entropía de la neurona VD a LP y viceversa, para los diferentes tamaños de palabras. Utilizando para ello la entropía normalizada.

Finalmente se propondrá otro tipo de representación de los eventos para calcular las probabilidades y así la IM en el sistema.

2 Cálculo de *spikes*

2.1 Lectura de los datos

2.1.1 Descripción

Para gestión de la información se utilizará la biblioteca de pandas, no es necesario gestionar la memoria con consideraciones adicionales porque las arquitecturas de nuestros ordenadores la manejan sin problemas.

La estructura de los ficheros viene dada en la información de los datos, en el fichero `InformacionFicheros.txt` y en las tres primeras líneas de los mismos (las cuales deberán de ser obviadas para la lectura del fichero).

2.1.2 Requisitos

- Tener las respectivas biblioteca instaladas (pandas, matplotlib, potly y numpy).
- Los datos deben encontrarse en el path indicado en la variable `data_path`.

En el siguiente fragmento de código puede observar la cabecera de los datos:

- El intervalo de muestreo es de $0.1ms$.
- Hay dos canales, una por cada neurona.
- Y en total se han tomado 19847700 muestreos.

```
print('Datos fichero trozo C')
print(23*'-')
!head -n 14 ./DatosSinapsisArtificial/InformacionFicheros.txt
```

Datos fichero trozo C

La estructura de los ficheros es:

TrozoC.txt -> Control
Las tres primeras lineas del fichero son:
Sample interval = 0,100000 ms
Number of channels = 2
Number of samples per channel = 19847700

y a continuación las columnas:
Columna 1 -> LP
Columna 2 -> VD

```
## Data information
sample_interval = 0.1
samples_per_channel_trozoC = 19847700
```

2.2 Lectura de los datos

Para la lectura de los datos se va a utilizar la biblioteca *Pandas* y la función `read_csv` puede encontrar la implementación de la misma en el directorio `src/read_data.py`.

```
from src.read_data import read_data, signal

signal['C'].head(4)
```

	LP	VD
0	0.004883	0.015259
1	0.001526	0.024109
2	-0.010681	0.031128
3	-0.022278	0.041809

Figura 1: Primeras 4 filas de la señal leída.

2.3 Diseño del algoritmo de cálculo de *spikes*

Para calcular los *spikes* se ha optado por utilizar un doble umbral. A continuación procedemos a detallar el algoritmo, puede encontrar la implementación en `src/signal_to_binary.py`:

Dada una señal `signal` que es una lista unidimensional de la señal. Se pretende detectar sus estímulos, para ello debe de superar el umbral superior `upper_threshold` y ser la primera vez o que ya se haya alcanzado un valor inferior a `lower_threshold`.

Además una vez que se supera el umbral se colocará cuando la tendencia vaya a bajar. Esto queda reflejado con los siguientes estados:

- **Estado 1:** Si `s > upper_threshold` entonces :

- i) `last = s`
- ii) pasar a estado 2.

- **Estado 2:** Si `s < last` entonces:

- i) poner un spike en señal anterior
- ii) `last = -inf`
- iii) pasar a estado 3 Si no entonces:
- iv) `last = s`

- **Estado 3:** Si `s < lower_threshold` entonces:

- i) Cambiar a estado 1

2.3.1 Motivación del algoritmo

Vamos a proceder a motivar el diseño del algoritmo, para visualizar mejor la explicación puede consultar (**Dualidad-spikes?**) .

Notemos que este algoritmo detecta el *spike* como el primer instante antes de que la señal empiece a decaer (punto azul) y en situaciones donde tras una caída no lo suficientemente baja y una subida aunque sea superior (punto amarillo) se tomaría al primero como punto de *spike*.

Esta decisión se ha tomado ya que filosóficamente se podría entender *spike* como el instante en el que toma un *valor grande* y que el resto son oscilaciones del pico. En caso de que se desee tener el valor amarillo bastaría con subir el umbral superior.

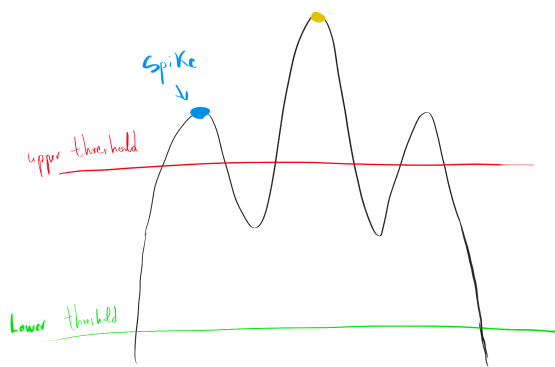


Figura 2.1: Dualidad spikes

2.4 Determinación de los umbrales

Para determinar los umbral vamos a calcular dos umbrales.

Bajo supuesto de que los valores que toma la señal siga una distribución normal; algo habitual, ya que este tipo de distribución modela fenómenos con mecanismos complejos y desconocidos.

Podemos tener presente de que esta distribución posee la propiedad de que podemos calcula el porcentaje de la población que hay bajo un punto a partir de la media y la desviación típica, en la Tabla 2.1 quedan recogidos algunas de las situaciones que vamos a utilizar.

Tabla 2.1: Desviaciones media con respecto a la desviación típica

Porcentaje de la población dentro de la normal	Distancia a la media
80%	1.281σ
90%	1.645σ
95%	1.956σ
99%	2.576σ
99.9%	3.291σ
99.99%	3.891σ
99.999%	4.892σ
99.9999%	5.326σ
99.99999%	6.109σ

Utilizaremos esta propiedad a nuestro favor con un doble objetivo:

- Obtener una idea intuitiva de dónde empezar a posicionar los umbrales por primera ver. Ya que los *spikes* o *impulsos* pueden entenderse como eventos *raros*.
- Una vez determinados los umbrales saber efectivamente cuánto de raros es el valor de un estímulo en cierta señal.

Hemos realizado por tanto un experimento para ver la dependencia entre el umbral seleccionado y el número de *spikes* detectados.

Éste consiste en variar los umbrales conforme a la distancia a media y ver el número el número de spikes detectados (puede consultar la implementación en `src/get_thresholds.py`)

El experimento puede ser ejecutado con `make experimento_umbrales` y se encuentra implementado en el fichero `src/experiment_gets_threshold.py`.

En una primera tentativa de calcular el umbral para el trozo C, neurona j LP los resultados han sido los siguientes (ver Tabla 2.2):

Tabla 2.2: Primera aproximación de los umbrales para neurona LP el trozo C

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.956	1.956	-0.161	0.161	54464
1.956	2.57	-0.161	0.211	41905
1.956	4.892	-0.161	0.402	31065
2.57	1.956	-0.211	0.161	45446
2.57	2.57	-0.211	0.211	38986
2.57	4.892	-0.211	0.402	31064
4.892	1.956	-0.402	0.161	10408
4.892	2.57	-0.402	0.211	10343
4.892	4.892	-0.402	0.402	10241

Se aprecia en Tabla 2.2 que el umbral bajo determina crucialmente el número de *spikes* realizaremos una inspección visual para ver qué está aconteciendo en varias secciones aleatorias de la muestra (si se encuentra en un entorno de ejecución podría modificar los valores de `higher_thresholds` y de `lower_threshold`).

Hemos repetido el experimento para cada uno de los trozos y cada neurona. Puede consultar los resultados en el apéndice 7.1 o bien ejecutarlos por si mismo `make experimento_umbrales`; ese comando mostrará los resultados en la terminal y además los almacenará en el directorio `experiment_results/get_threshold.txt`.

A la vista de los resultados de los primeros umbrales en un primer estadio hemos tomado como criterio tener los umbrales lo más *grandes* posibles siempre y cuando el número de spikes no decaiga dramáticamente. La selección primera ha resultado:

Tabla 2.3: Selección primera de umbrales

Trozo	Neurona	Umbral inferior	Umbral superior	Número de spikes
C	LP	-0.320	0.402	30308
C	VD	-0.084	0.205	21246
R	LP	-0.534	0.920	24076
R	VD	-0.085	0.206	17618
G	LP	-0.316	0.397	25889
G	VD	-0.120	0.207	13127

Como método de validación de estos umbrales hemos formulado el siguiente experimento:

Para cada trozo y neurona se realizará una inspección visual de cinco rango aleatorios de valores, si para estos se escapan *impulsos* que visualmente se consideran válidos se retocará el umbral.

Puede ejecutar el experimento con: `make plot_experimental_thresholds` que no solo mostrará en pantalla las gráficas si no que las almacena en la carpeta `img/04_calculo_spikes`.

Vamos a proceder a mostrar algunos de los ejemplos representativos:

Para la señal C neurona LP va a ser necesario subir un poco el umbral inferior, ya que en dos casos *spikes* de rangos aleatorios se ha escapado un estímulo por el rango inferior como se puede apreciar en Figura 2.2.

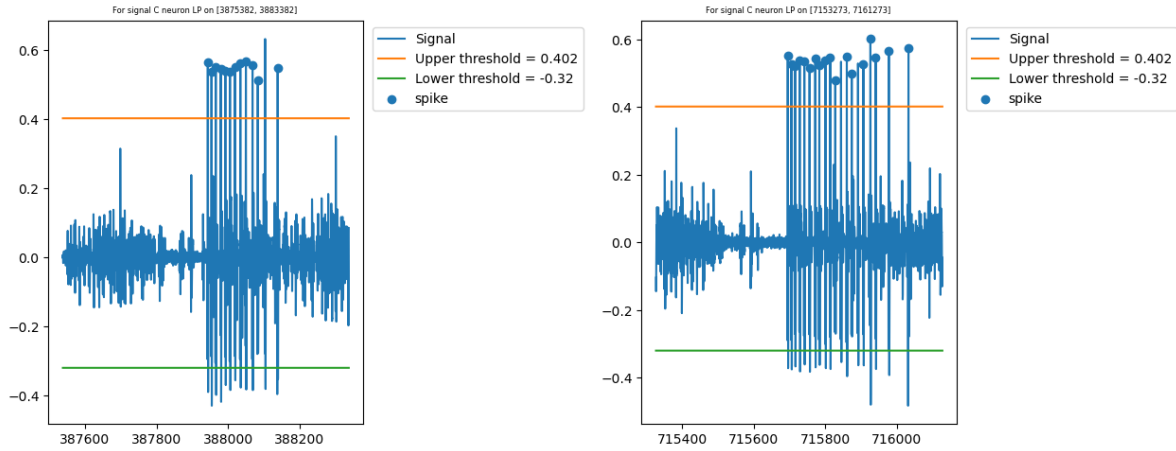


Figura 2.2: Umbral inferior demasiado bajo para señal C, neurona LP

Es por ello que vamos a considerar el nuevo umbral bajo como -0.211.

Para la misma señal la neurona VD ocurre un efecto parecido que procederemos a paliar aumentando el umbral (Figura 2.3).

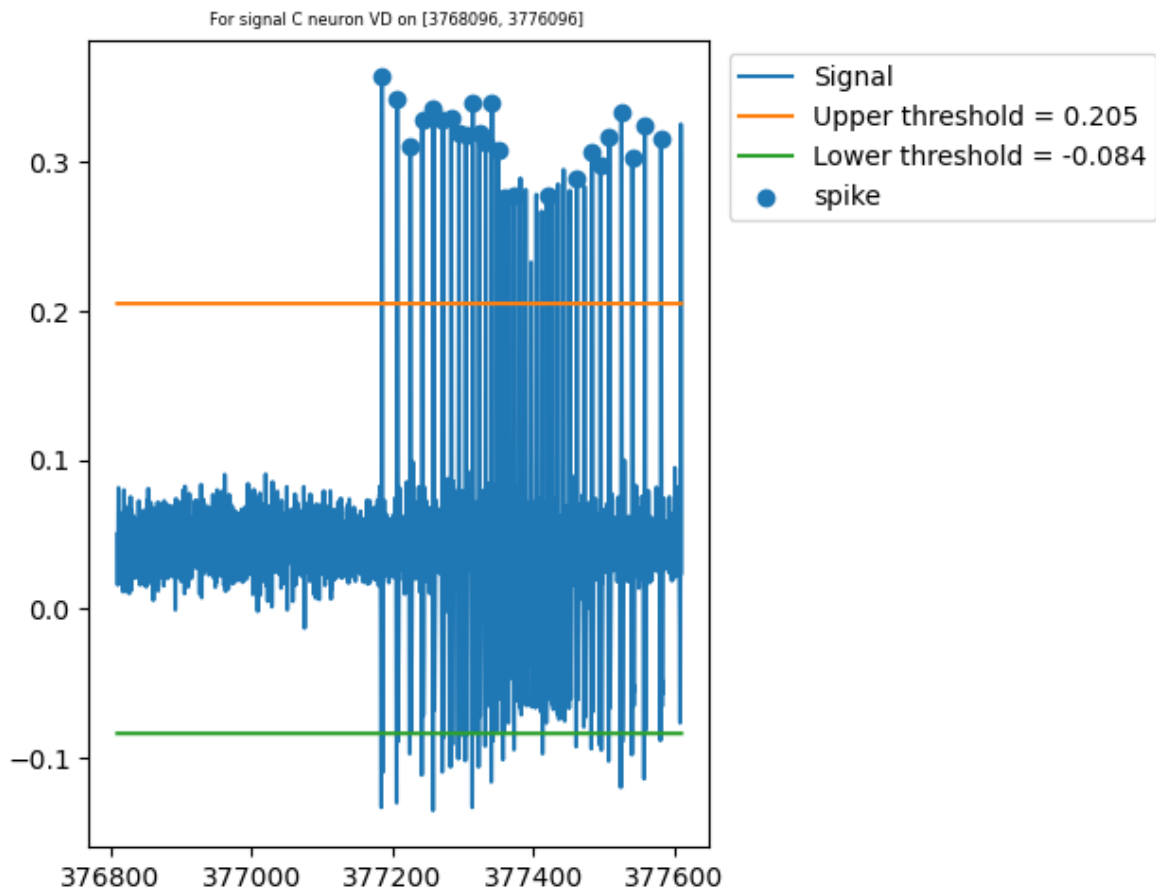


Figura 2.3: Umbral inferior demasiado bajo para señal , neurona VD

Es por ello que lo subiremos a -0.041 .

Para la señal R de la neurona LP los *spikes* se separan demasiado de los umbrales (ver Figura 2.4).

Vamos a ajustar los umbrales a los que ahora se aprecian en Figura 2.6

Para el caso de la misma señal neurona VD (ver Figura 2.8), podemos observar que es el umbral es correcto.

Para G LP los umbrales están demasiado cerca de la media (ver Figura 2.7).

Tras estas observaciones puede se ha concluido que

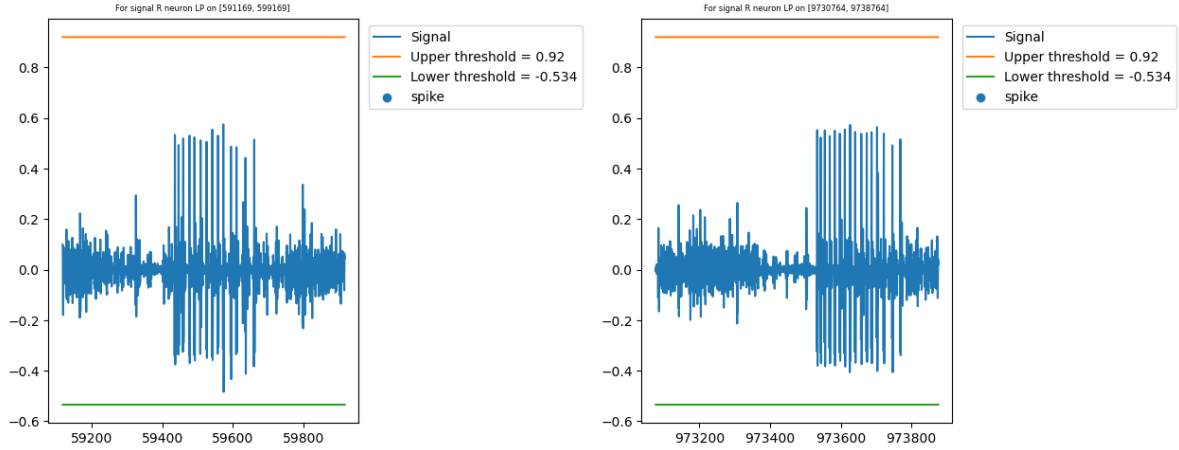


Figura 2.4: Umbrales demasiado alejados de la señal para la señal R neurona LP.

Tabla 2.4: Selección final de umbrales

Trozo	Neurona	Distancia σ inferior	σ superior	Umbral inferior	Umbral superior
C	LP	2.57	4.892	-0.211	0.402
C	VD	2.57	3.891	-0.041	0.172
R	LP	1.4	1.956	-0.306	0.348
R	VD	3.891	4.892	-0.085	0.206
G	LP	5.326	> 15	-0.432	0.75
G	VD	3.891	4.892	-0.087	0.207

Por la determinación de estos umbrales puede observarse cierta asimetría de la señal: *sube más de lo que baja*, también es llamativo que para una misma neurona cada umbral varíe, esto puede deberse o a la naturaleza propia de la señal o que el aparato de medida o condiciones del experimento sean distintas.

Puede observar las gráficas finales en la carpeta `img/04_calculo_spikes` las que comiencen con 2 o bien ejecutando usted mismo `make plot_experimental_thresholds` (o directamente el programa `python src/experiment_view_threshold.py`).

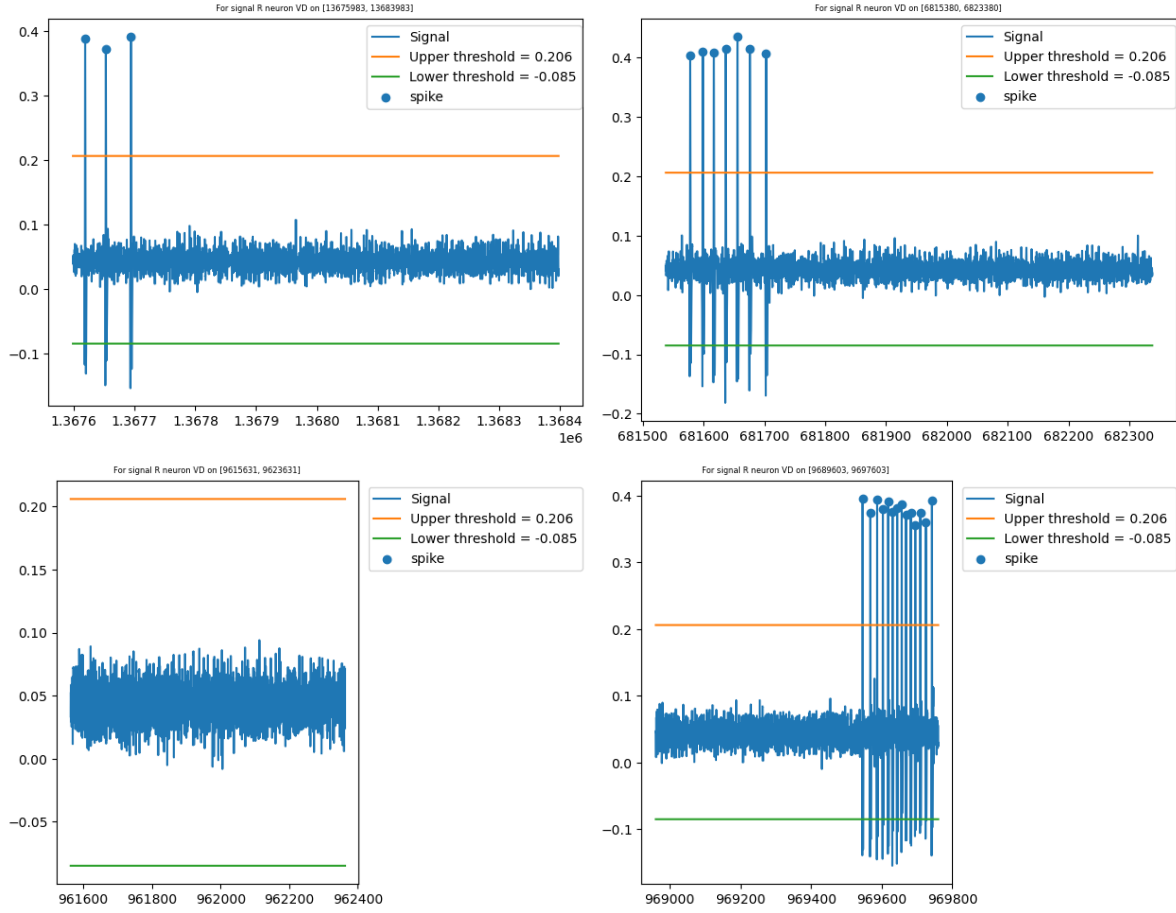


Figura 2.5: Región sin *spike*

Figura 2.6: Umbrales correctos para R neurona VD.

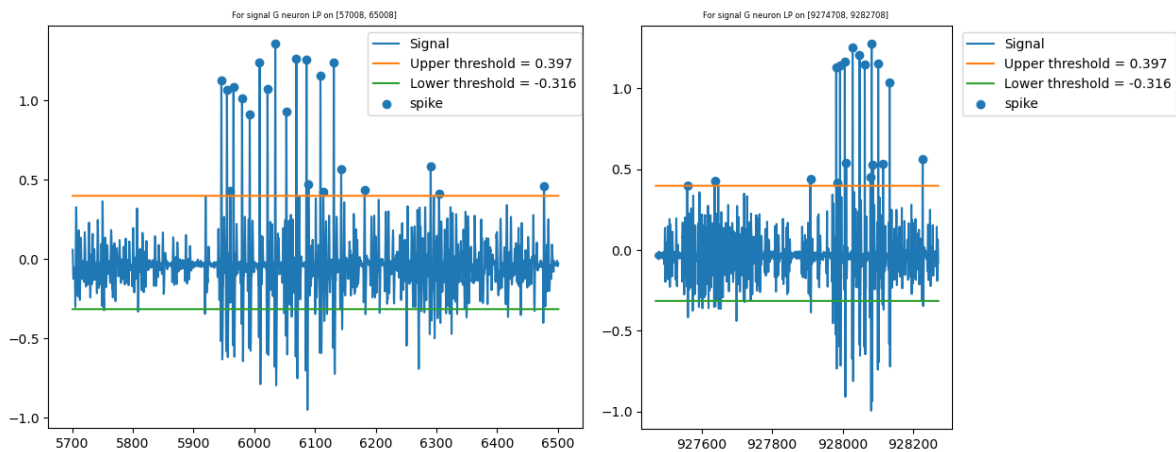


Figura 2.7: El umbral superior está demasiado cerca

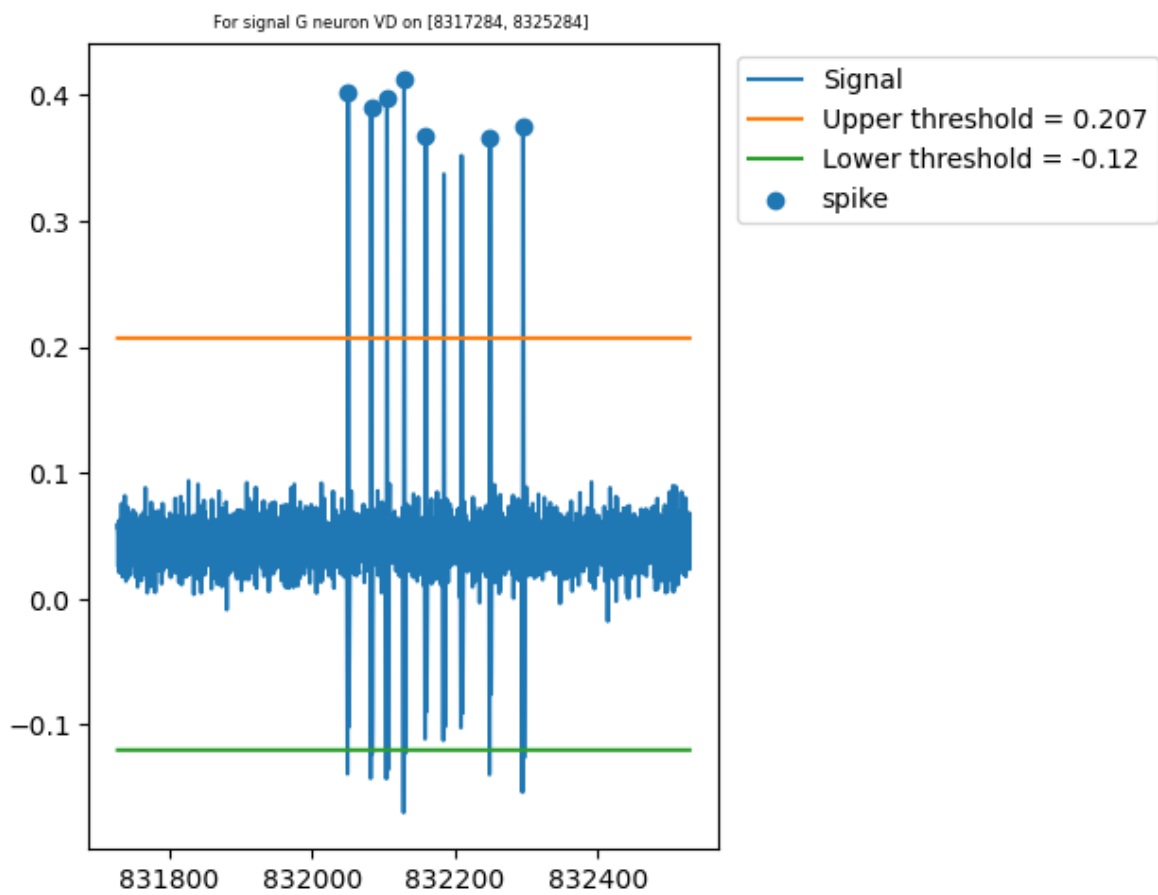


Figura 2.8: Para la señal G neurona VD el umbral inferior es demasiado bajo.

3 Cálculo de la información mutua

Vamos a proceder con el cálculo de la información mutua entre las dos señales.

3.1 Abstracción del problema

Una vez preprocesada la señal a una secuencia binaria, donde uno significa hay estímulo; la probabilidad de que aparezca una determinada palabra (cadena de n -bits consecutivos) se trata de un proceso estocástico de Poisson.

Es decir, la probabilidad de que pueda aparecer cierta cadena.

El estimador máximo verosímil de una distribución de Poisson es la media, luego fijado un tamaño de palabra y un *stride* (deslizamiento de la ventana) calcularemos la frecuencia de cada casuística y esa será su probabilidad.

3.2 Cálculo de la información mutua

La información mutua para dos variables X, Y aleatorias se puede definir como

$$MI(X, Y) = S(X) + S(Y) - S(X, Y), \quad (3.1)$$

donde S denota a la entropía y responde a las siguientes fórmulas:

$$S(X) = - \sum_i p(x_i) \log_2(p(x_i)) \quad (3.2)$$

Las implementaciones de estas funciones se encuentran en el ejecutable `src/formulas.py`.

3.3 Formulación del experimento

Para el cálculo de la información mutua se va a realizar el siguiente proceso:

1. Se transforma la señal analógica en una binaria (ver algoritmo `src/signal_to_binary.py`).
2. Se fija un tamaño de ventana y *stride*.
3. Para la ventana y *stride* de las señales *X* e *Y* se obtienen un array de sus palabras.
4. Se calcula la probabilidad de tales

A la vista de los resultados de cada trozo podemos afirmar que el *stride* no juega un papel fundamental pero que sí lo hace el tamaño de ventana.

Es por ello que vamos a fijar de ahora en adelante *stride* = *bits* y vamos a aumentar el ancho de ventana, tomaremos las siguientes casuísticas y las compararemos a posteriori:

La condición de tamaño de ventana:

- No genera ninguna ventana falsa.
- Cuantil 0.05 de las distancias entre *spikes*.
- Aceptaremos un 5% de ventanas falsas.

La implementación del cálculo de ventana se realiza en `/src/spike_distance.py` puede ejecutar tal fichero o hacer `make calcular_distancia_spike` el código se segmenta en:

- Calcular las distancias entre *spikes* consecutivos.
- Calcular la distancia mínima.
- Calcular el cuantil 0.05 de las distancias calculadas.
- Realizar una búsqueda binaria del tamaño de ventana que hace que al rededor del 5% sean ventanas falsas.

Los resultados han sido los recogidos en Tabla 3.1:

Tabla 3.1: Mínima distancia entre los *spikes*

Trozo	Neurona	Intervalo mínimo entre <i>spikes</i>	Media distancias	std
C	LP	24	9994824.93	6056023.94

Trozo	Neurona	Intervalo mínimo entre spikes	Media distancias	std
C	VD	10	10229353.49	5779379.91
R	LP	16	8356519.06	4859624.69
R	VD	16	8422905.58	4842002.99
G	LP	17086	9173668.30	4109764.72
G	VD	65	8413845.72	4894766.83

Tabla 3.2: Distancia de *spike* para cuantil de distancia de 0.05

Trozo	Neurona	Distancia cuantil 0.05
C	LP	108
C	VD	86
R	LP	41
R	VD	96
G	LP	26336
G	VD	92

Tabla 3.3: Tamaño de ventana en función de un porcentaje de ventanas falsas

Trozo	Neurona	Porcentaje ventanas falsas	Distancia admitir ventanas falsas
C	LP	4.91%	186
C	VD	4.94%	207
R	LP	5.01%	124
R	VD	4.92%	213
G	LP	4.72%	153362
G	VD	5.02%	247

Es por ello que ampliaremos el tamaño de ventana es decir transformaremos la señal binaria de cada trozo. Ahora habrá un máximo de 5% de ventanas falsas.

Donde el tamaño de ventana nuevo, para cada trozo vendrá dado como:

$$\text{tamaño ventana} = \min(\text{distLP}, \text{distVD}) + 1$$

Tabla 3.4: Ventanas máximas calculadas

Trozo	Ventana máxima distancia	Ventana máxima cuantil 0.5	Ventana máxima falsas
C	11	87	187
R	17	42	125
G	66	93	248

Las nuevas señales binarias vendrán dadas por:

$$\text{señal binaria nueva}[i] = \max \text{señal antigua}[i * \text{radio acción} : (i+1) * \text{radio acción}]$$

De esta manera la señal tendrá un uno si ya lo había o un dos si no lo había.

Mostraremos la información mutua calculada en la sección siguiente.

4 Cálculo de entropía normaliza o información mutua normalizada

Esta medida se usa para medir la transferencia de información del estímulo S a la neurona respuesta R .

Viene dada por la siguiente expresión

$$E_{RS} = \frac{MI_{RS}}{H(S)}$$

Donde H ya hemos comentado que es la entropía de S , es decir la máxima cantidad de información que se puede transmitir del estímulo a la neurona respuesta.

Está acotado entre

$$0 \leq E_{RS} \leq 1$$

Si $E_{RS} = 0$ significa que toda la información es perdida, es decir respuesta y estímulo son completamente independientes. $E_{RS} = 1$ sería la sincronización completa.

4.1 Resultados obtenidos

4.1.1 Trozo C

trozo	window size	bits	stride	IM(LP,VD)	E(LP -> VD)	E(VD -> LP)
C	11	1	1	0.00026	0.00276	0.00212
C	11	2	2	0.00106	0.0055	0.0042
C	11	3	3	0.0024	0.0084	0.0064
C	11	4	4	0.0043	0.0113	0.0087
C	11	5	5	0.0069	0.01446	0.011
C	11	6	6	0.0102	0.0178	0.0137
C	11	7	7	0.0143	0.021	0.01655

trozo	window size	bits	stride	IM(LP,VD)	E(LP -> VD)	E(VD -> LP)
C	11	8	8	0.0189	0.0248	0.0190
C	87	1	1	0.01839	0.0399	0.0320
C	87	2	2	0.0617	0.068	0.0544
C	87	3	3	0.0825	0.0686	0.054
C	87	4	4	0.0970	0.066	0.0524
C	87	5	5	0.1105	0.06	0.0513
C	87	6	6	0.126	0.066	0.0514
C	87	7	7	0.144	0.068	0.052
C	87	8	8	0.1618	0.0696	0.054
C	187	1	1	0.0657	0.0989	0.082
C	187	2	2	0.0973	0.097	0.08
C	187	3	3	0.119	0.092	0.0764
C	187	4	4	0.1424	0.090	0.075
C	187	5	5	0.17	0.0938	0.077
C	187	6	6	0.211	0.0998	0.083
C	187	7	7	0.264	0.111	0.092
C	187	8	8	0.328	0.124	0.104

4.1.2 Trozo R

trozo	window	size	bits	stride	IM(LP,VD)	E(LP -> VD)	E(VD -> LP)
:-:	:-:	:-:	:-:	:-:	:-:	:-:	:-:
G	66	1	1	1.620257553269777e-05	4.788598235288777e-05	0.006711196664509015	
G	66	2	2	6.418918082318203e-05	9.488712855603435e-05	0.013312216678317107	
G	66	3	3	0.00013099340609190602	0.0001329391500226142	0.01809334912961612	
G	66	4	4	0.00020205648709792356	0.0001628170704016028	0.021059375964410595	
G	66	5	5	0.00027275212246169467	0.0001876460997011146	0.022732554047891478	
G	66	6	6	0.0003462228201733808	0.00021049636667214128	0.024005795801305914	
G	66	7	7	0.0004245960074780175	0.00023191175122932325	0.025261930551116026	
G	66	8	8	0.000504442560984586	0.0002516244975164341	0.026465933558880647	
G	93	1	1	3.244502013038941e-05	7.548964708043917e-05	0.009888357901355609	
G	93	2	2	0.00011832752280804826	0.00014016709846578274	0.018095996281303334	
G	93	3	3	0.00021767397895899876	0.0001865880390852357	0.022172974382483238	
G	93	4	4	0.0003197021433145597	0.00022411898588438014	0.024455706467387005	
G	93	5	5	0.00042712311687020943	0.0002567665675718544	0.02620042840149757	
G	93	6	6	0.0005409884780842547	0.00028767481807033724	0.02766487880048169	
G	93	7	7	Col 6: IM(LP,VD) ,124	0.00031747751318104273	0.029476262147995928	
G	93	8	8	0.0007978512356459078	0.0003440192776831993	0.030564401869198097	
G	248	1	1	0.00018445144639545052	0.0002727347185191284	0.023561436223382873	
G	248	2	2	0.0004625940428346631	0.00044338925557154904	0.029593792210503005	
G	248	3	3	0.0007978512356445755	0.0005742888697831222	0.034085803864510905	
G	248	4	4	0.0012025720124806405	0.0006981880189705235	0.03878950209350325	
G	248	5	5	0.0016858652991103007	0.000820392934364245	0.043333967673638354	
G	248	6	6	0.0022197247303252787	0.000936842125018898	0.04772697706464793	
G	248	7	7	0.002909173447635638	0.001080472414534834	0.05356450317715368	
G	248	8	8	0.0038651410884460446	0.001289684417378414	0.06248336821500431	

...

...

4.1.3 Trozo G

	trozo	window	size	bits	stride	IM(LP,VD)	E(LP -> VD)	E(VD -> LP)
	:-:	:-:	:-:	:-:	:-:	:-:	:-:	
	R	17	1	1	6.685320383392757e-07	5.077194403185326e-06	2.197365939251354e-06	
	R	17	2	2	9.57249589439435e-06	3.641102887197067e-05	1.5803053136198835e-05	
	R	17	3	3	7.335448171374992e-05	0.00018632227056256692	8.086644898114785e-05	
	R	17	4	4	0.00023656758893020324	0.0004514811943632702	0.00019585023609371732	
	R	17	5	5	0.0004968064271215766	0.0007596826654380687	0.0003292708660247043	
	R	17	6	6	0.0011597112170957935	0.0014794518526010107	0.0006411550088840745	
	R	17	7	7	0.0024338804989487706	0.00266725765661794	0.0011539775890038984	
	R	17	8	8	0.005738647256177565	0.005524014379004146	0.002383784263940942	
	R	42	1	1	8.356785525576171e-06	3.148550945182821e-05	1.4792386425803575e-05	
	R	42	2	2	7.75743884839919e-05	0.00014685827248318254	6.865757850477081e-05	
	R	42	3	3	0.0003651094481629258	0.00046207815614229575	0.0002154687315771534	
	R	42	4	4	0.0034725962905142183	0.003375396837181007	0.0015479330039863196	
	R	42	5	5	0.011873643696266534	0.009665310806548599	0.004283630080925182	
	R	42	6	6	0.027986810748303448	0.019924942449643636	0.008492935398260886	
	R	42	7	7	0.054578425232929106	0.03487372511577423	0.014311156083233868	
	R	42	8	8	0.0949590717798996	0.05523983451852838	0.021962162954755688	
	R	125	1	1	8.371207177670925e-05	0.00014981536571265585	9.004879477215312e-05	
	R	125	2	2	0.011283248240873345	0.011394919599949326	0.00628502884560713	
	R	125	3	3	0.027942427574982	0.021462194314631738	0.010669125620828165	
	R	125	4	4	0.05981403191906853	0.037697625885814504	0.01750212579102524	
	R	125	5	5	0.10881967586651164	0.058320285792779	0.025886085619388422	
	R	125	6	6	0.2241450568400598	0.10500569451067515	0.044927475265202466	
	R	125	7	7	0.4449355759022984	0.1855814843291307	0.0772662239344192	
	R	125	8	8	0.7452863675983021	0.2797915850670172	0.11442711631470885	

...
...

5 Suposición de otro tipo de codificación de eventos

Se pretende en este apartado utilizar otro tipo de codificación de eventos para el cálculo de las probabilidades y la información mutua.

En nuestro caso vamos a proponer el sistema de codificación SAX (ver artículo Lin et al. (2003)).

5.1 Descripción del sistema de codificación SAX

SAX permite reducir una serie temporal de tamaño n a otra de tamaño w usualmente $w \ll n$.

El tamaño del alfabeto, a , será con la restricción de que $a > 2$.

Los pasos a seguir son:

1. Transformación de los datos en **PAA** *Piecewise Aggregate Approximation*

De la siguiente manera: Una serie temporal C de longitud n puede ser representada en un vector \bar{C} de dimensión w .

El elemento i -ésimo de \bar{C} viene dado por

$$\bar{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j.$$

Notemos que esto no es más que una media de los elementos contiguos.

2. **Discretización**

Cada componente de la nueva señal transformada será mapeada por un símbolo en función del rango de valor en que se encuentre.

Para ello se definen los *breakpoints*, que no son más que una lista de números $B = b_1, \dots, b_{a-1}$ de tal forma que su area bajo un normal $\mathcal{N}(0, 1)$ sea para cada uno de ellos $\frac{1}{a}$. Además se tiene que $b_0 = -\infty$ y $b_a = +\infty$.

De esta manera formaremos las palabras \hat{C} que vendrá dada como

$$\hat{c}_i = \alpha_j, \text{ sii } \beta_{j-1} \leq \beta_j$$

y con esto se obtendría la nueva señal.

5.2 Observaciones

Notemos que se tienen dos variables libres en este sistema:

- w el tamaño de PAA, que de manera implícita debe de ser un divisor de n el tamaño original para mayor comodidad.
- a el tamaño del alfabeto.

5.3 Sobre nuestra implementación

Puede encontrar la implementación de los algoritmos descritos en `src/SAX.py`, en ellos se encuentran fielmente escritos los pasos mencionados.

Cabe destacar que se han tomado las siguientes decisiones en diseño:

- Por eficiencia y poder reutilizar el código de información mutua se ha tomado por alfabeto a un subconjunto de tamaño a de los números reales.

5.4 Experimentación con SAX

5.4.1 Descripción del experimento

Pretendemos calcular la información mutua en distintas circunstanancias y compararlas con los datos anteriores.

Con este fin w el tamaño de la nueva señal de un trozo T vendrá dado por

$$w = \frac{\text{Tamaño de la señal del trozo } T}{w'}$$

donde w' es el tamaño de palabra que se usó en el trozo T de los apartados anteriores.

Por limitaciones computacionales se han seleccionado las siguientes combinaciones de prueba:

Tabla 5.1: Combinaciones de tamaño de palabra w' para SAX que se van a realizar

Trozo	Tamaños w'
C	11,87,187
R	17,42,125
G	66, 93,248

Tamaños de alfabeto: 1, 2, 5, 10, 20 y 50 independientemente del trozo que sea.

Tamaños de bits con el que calcular la información mutua: 1, 2, 3, 4, 5, 6, 7 y 8 independientemente del trozo que sea.

Para el experimento se han realizado todas las combinaciones posibles de estos parámetros y se ha calculado la información mutua pertinente.

Puede ejecutar el experimento realizando `make calcular_sax` o directamente escribiendo `python src/experiment_sax_mi.py`.

5.4.2 Resultados obtenidos

5.4.3 Trozo C

Puede encontrar los resultados completos en el apéndice sección 7.2.1.

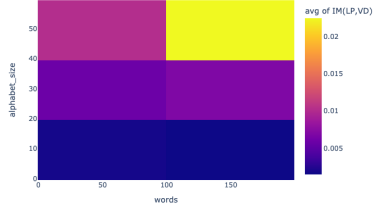
Una representación gráfica de los mismos con un mapa de color sería la mostrada en la figura Figura 5.1; para su interpretación tenga presente que a más claro mayor es la entropía.

A la vista de los resultados uno se da cuenta que por lo general aumentar el ancho de ventana mejora los resultados, pero que llegado a cierto punto va en detrimento.

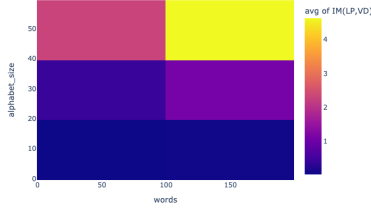
El ancho de ventana en este problema se puede ampliar en dos aspectos y tiene diferentes matices:

- **Tamaño de palabra para el cálculo de la información mutua**, lo que nosotros hemos denominado durante todo el experimento como bits b . Tiene en cuenta eventos temporales situaciones contiguas de tiempo.
- **Tamaño de palabra de reducción**, esto es lo que nosotros hemos denotado como w' , reduce a un símbolo la media de valores de esa palabra.

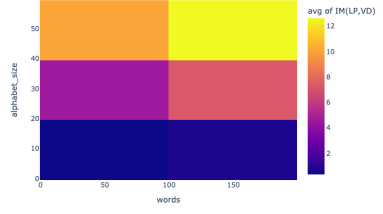
IM en función del tamaño de palabra y del alfabeto para 1 bits



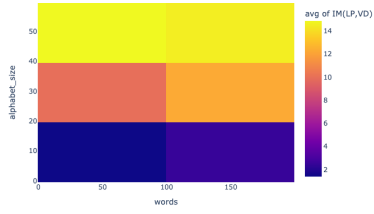
IM en función del tamaño de palabra y del alfabeto para 2 bits



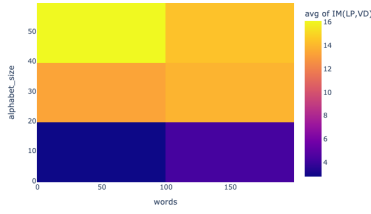
IM en función del tamaño de palabra y del alfabeto para 3 bits



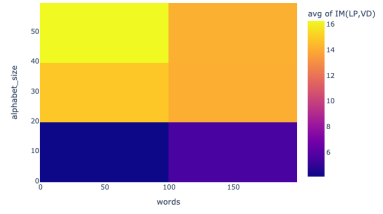
IM en función del tamaño de palabra y del alfabeto para 4 bits



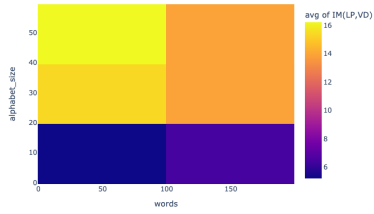
IM en función del tamaño de palabra y del alfabeto para 5 bits



IM en función del tamaño de palabra y del alfabeto para 6 bits



IM en función del tamaño de palabra y del alfabeto para 7 bits



IM en función del tamaño de palabra y del alfabeto para 8 bits

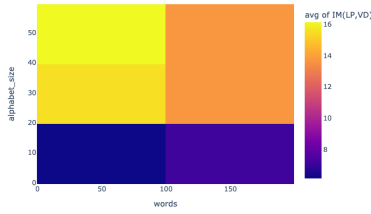


Figura 5.1: Mapa de color de la información mutua variando en función del número de bits, tamaño de palabra w' y tamaño de alfabeto a .

Una moraleja que se obtiene de aquí es que **aumentar el número de bits no significa mayor cantidad de información mutua**. Fijadas la palabra y tamaño de alfabeto, si bien al hacer crecer el número de bits se aprecia una tendencia creciente de la información mutua, existen ciertas excepciones.

Como muestra de ello se tienen las siguiente combinaciones, para las que aumentar el número de bits no implica una mejora en la información mutua:

Tabla 5.2: Tamaño de palabra y tamaño de alfabeto para los cuales la IM no aumenta con el número de bits

words	alphabet size
87	20
187	20
87	50
187	50

Queda reflejado su valor en la Figura 5.2

El máximo de IM se alcanza para 8 bits, con un tamaño de palabra 11 y tamaño de alfabeto 50.

Si ordenamos los parámetros por su eficiencia mutua puede encontrar a los diez primeros en la Tabla 5.3 .

Tabla 5.3: Combinaciones de información mutua máxima

bits	words	alphabet_size	IM(LP,VD)
8	11	50	17.477040
7	11	50	17.464702
6	11	50	17.349276
5	11	50	16.866878
8	11	20	16.268008
7	11	20	15.786441
5	87	50	15.349300
6	87	50	15.183218
4	87	50	15.093791
6	87	20	15.058446

A la vista de estos resultados podemos observar que para tamaños de bit lo suficientemente grande los parámetros clave han sido el tamaño de palabra 11 y un alfabeto de tamaño 50.

La tendencia a mejorar con el tamaño del alfabeto puede observarse gráficamente en la Figu-

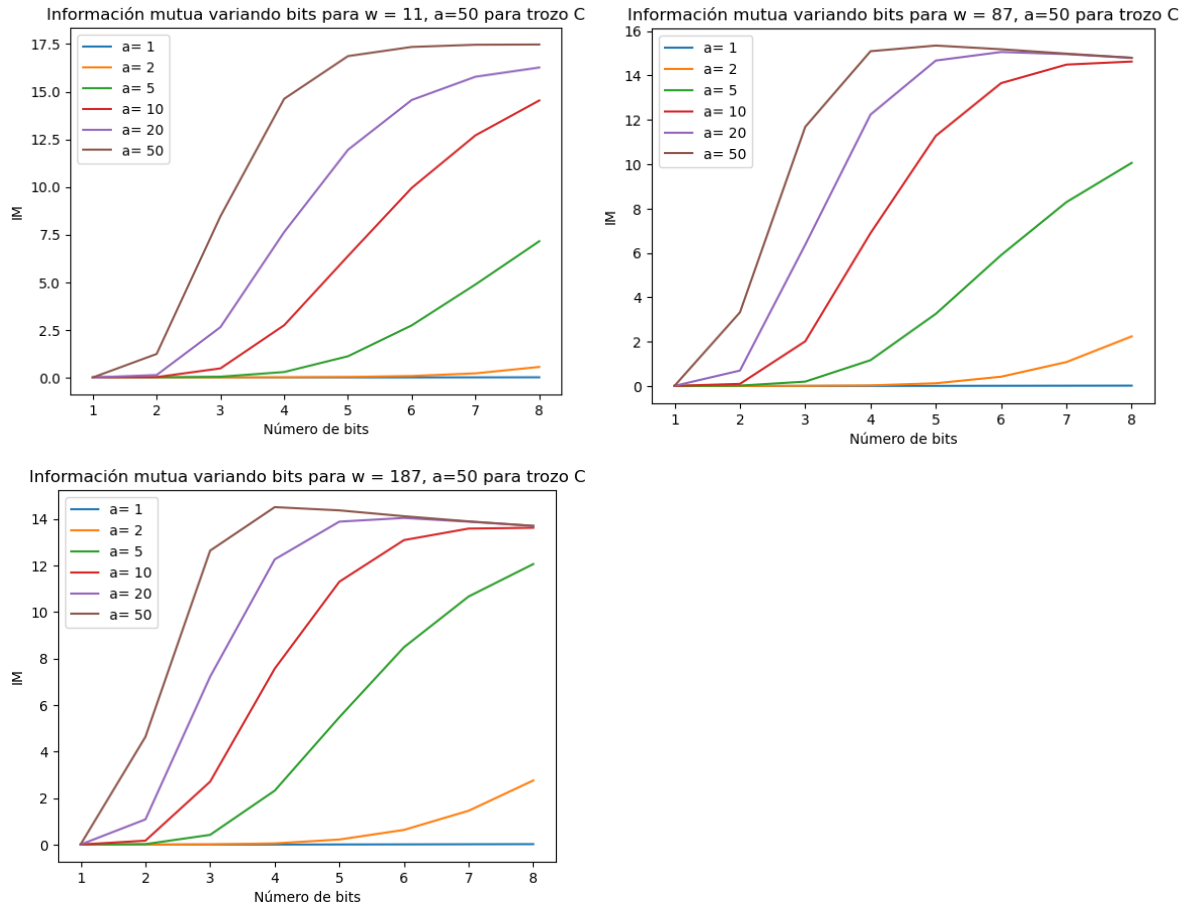


Figura 5.2: Información mutua variando bits para para trozo C

ra 5.3 :

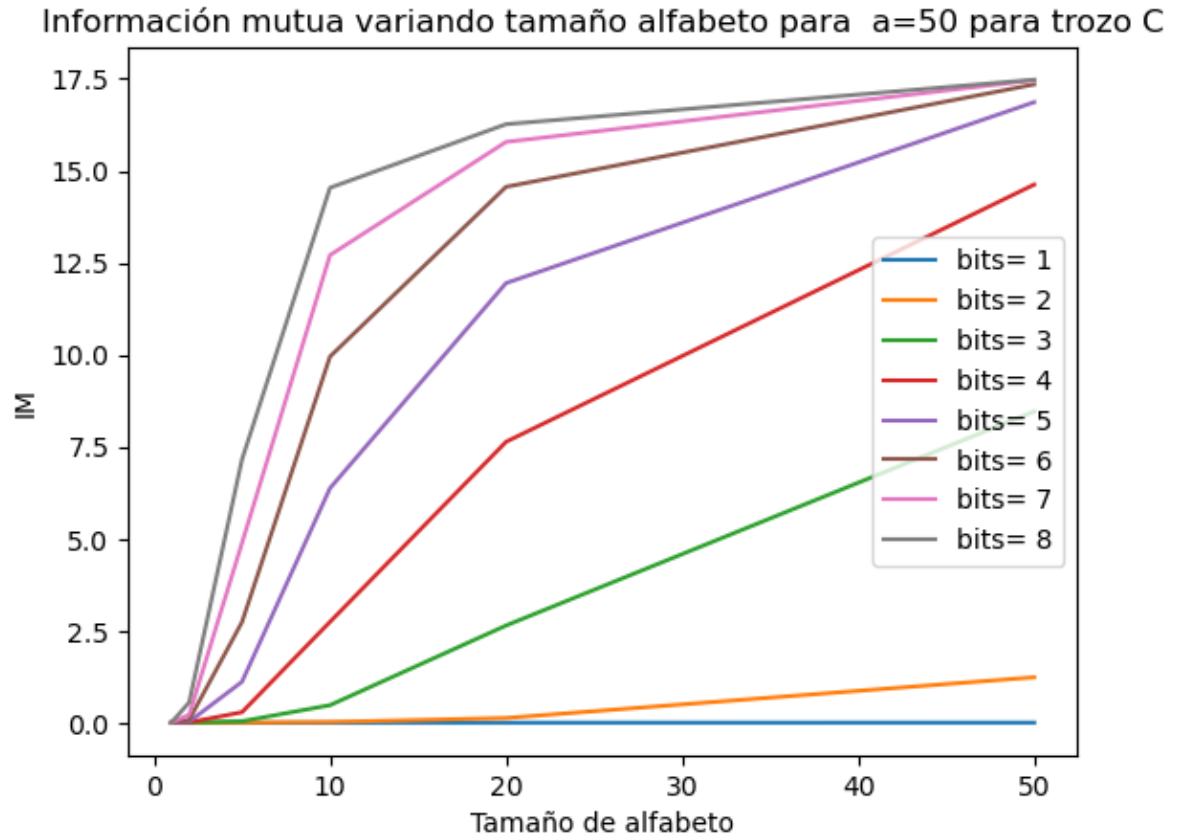
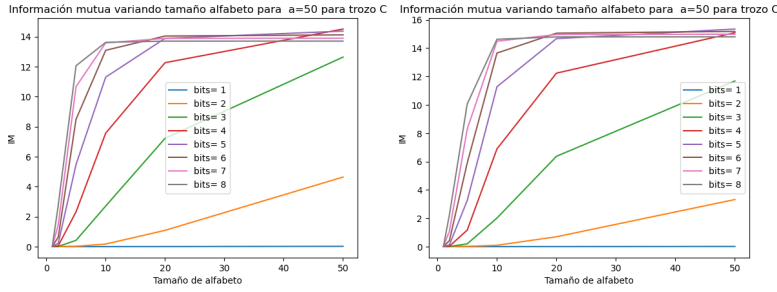


Figura 5.3:

A continuación, en la Figura 5.4 se muestra cómo influye el tamaño de palabra

Las conclusiones que se sacan es que cuando muchos *samples* se ven involucrados *la información se diluye* y por tanto la entropía baja. Como era de esperar, al añadir más símbolos la información mutua mejora en comparación con la primera codificación, la idea intuitiva de esto subyace en que tenemos *más pistas* de lo que va a ocurrir.

Un problema que tiene es que no contempla aportar la información para los spikes.

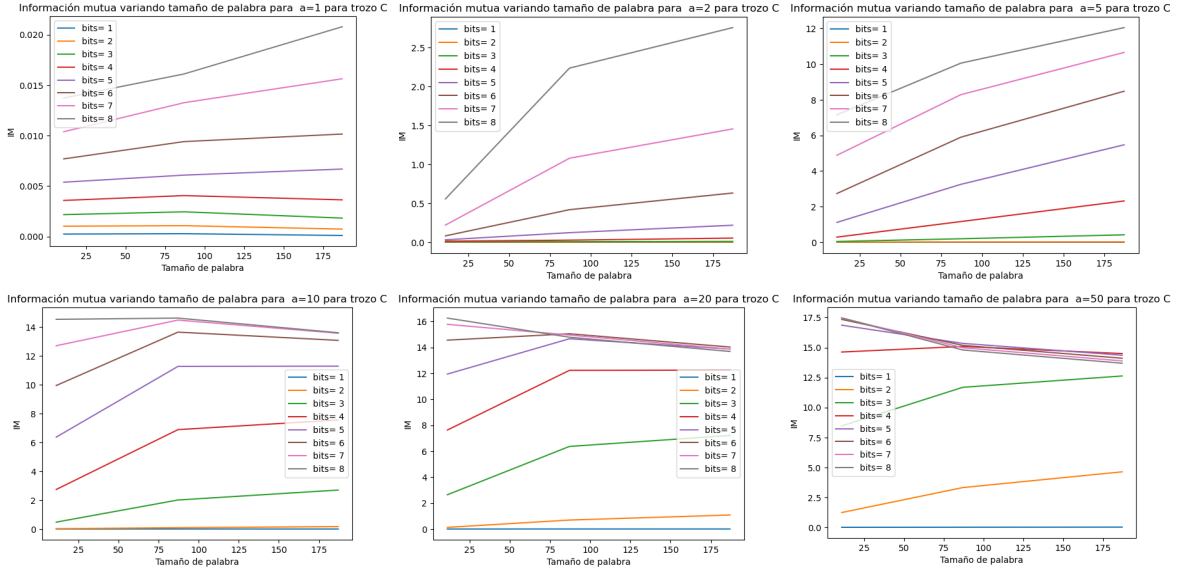


Figura 5.4: Información mutua variando bits para para trozo C

5.5 Comparativa

5.5.1 Ventajas

- La **reducción de tamaño** de la señal permite un tratamiento más rápido.
- Discretización de los valores que puede tomar la señal, esto simplifica su tratamiento.

5.5.2 Desventajas

- Pérdida de información.
- Rigidez a la hora de calcular \bar{C} ya que solo es el promedio de los datos de una palabra.
- Selección de hiperparámetros libre, se desconoce un método eficaz y riguroso como criterio de selección o descarte de los hiperparámetros a y w .

5.6 Trozo R

Se ha repetido el mismo análisis para el resto de trozos R y G, las conclusiones han sido equivalentes para estos dos últimos.

Puede encontrar los resultados numéricos del trozo R en el apéndice 7.2.2 (o en la carpeta `experiment_results/SAX`).

Puede contemplar el mapa de calor de la información mutua en la Figura 5.5

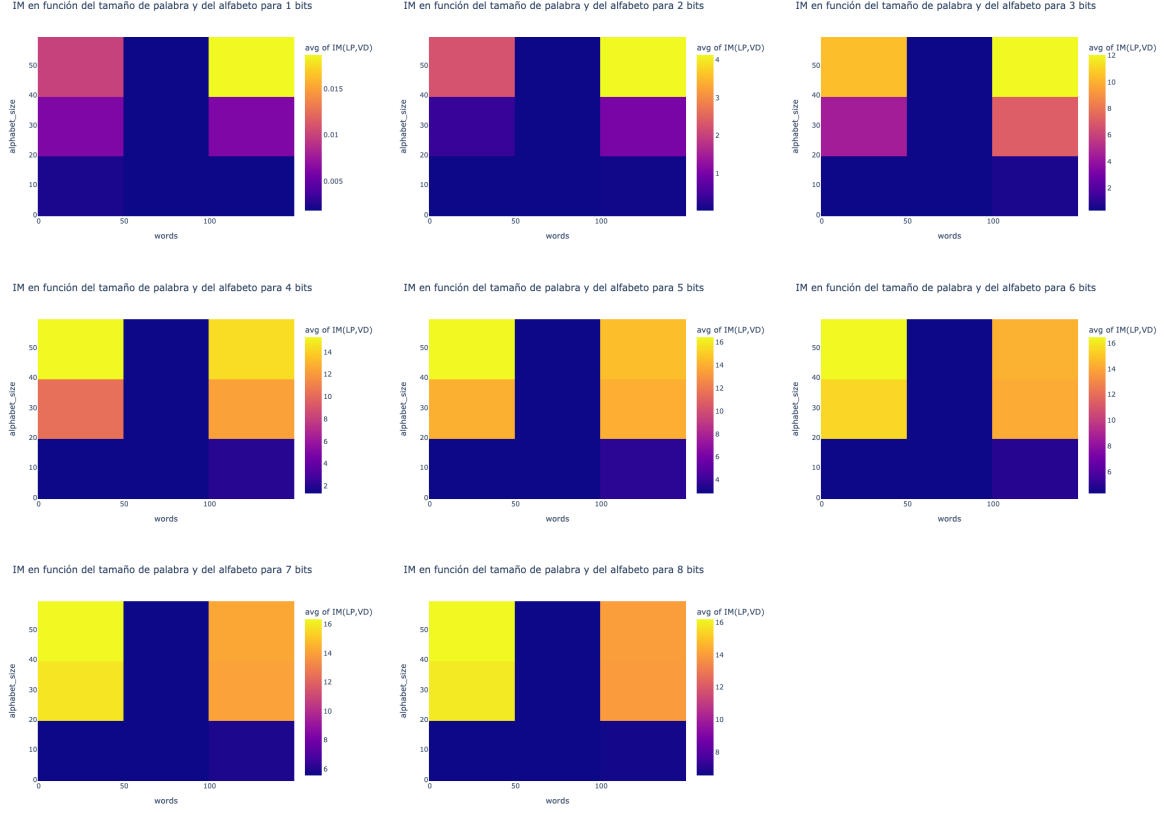


Figura 5.5: Mapa de color de la información mutua variando en función del número de bits, tamaño de palabra w' y tamaño de alfabeto a .

Los diez mejores valore de información mutua han sido

Tabla 5.4: Combinaciones de información mutua máxima para el trozo R

bits	words	alphabet_size	IM(LP,VD)
6	17	5	6.977234
7	17	5	6.923350
8	17	5	6.813161
5	17	5	6.790366
8	17	2	6.218410
7	17	2	6.061772
5	42	5	6.053265
6	42	5	5.940968
7	42	5	5.754503

bits	words	alphabet_size	IM(LP,VD)
7	42	2	5.612413

5.7 Trozo G

Puede encontrar los resultados numéricos del trozo G en el apéndice 7.2.3 (o en la carpeta `experiment_results/SAX`).

Puede contemplar el mapa de calor de la información mutua en la Figura 5.6 , preste atención a los valores del rango con cuidado.

Tabla 5.5: Combinaciones de información mutua máxima para el trozo G

bits	words	alphabet_size	IM(LP,VD)
5	66	50	15.460422
6	66	5	5.318634
6	66	2	5.180586
7	66	5	5.112652
7	66	2	5.077938
5	93	5	5.059765
8	66	5	4.921054
8	66	2	4.911715
4	66	5	4.905467
4	93	5	4.890374

5.8 Conclusiones trozo C y G

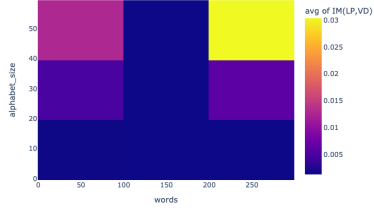
TODO

5.9 Futuros campos que contemplar esta codificación

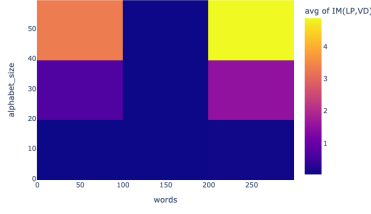
A la hora de presentar el problema se indica que el tamaño final de la cadena es w , sería interesante reenfocarlo como nosotros hemos hecho, en vez de que la media se de los w_0/w definir la variable w' que sería el número de elementos sobre los que se haría la media.

La longitud por tanto vendría dada como w_0/w' y lo que es más interesante se podría jugar con el *stride* (desplazamiento de ventana) y hacer de la media una pondera.

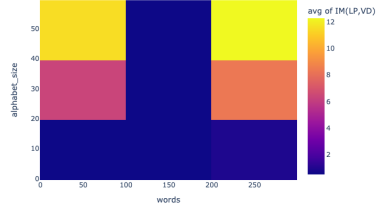
IM en función del tamaño de palabra y del alfabeto para 1 bits



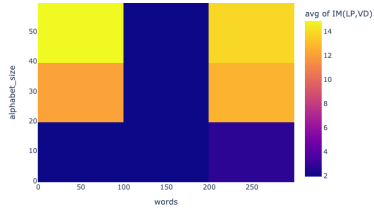
IM en función del tamaño de palabra y del alfabeto para 2 bits



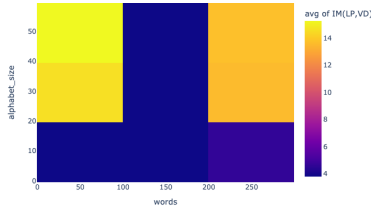
IM en función del tamaño de palabra y del alfabeto para 3 bits



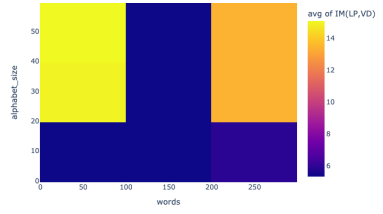
IM en función del tamaño de palabra y del alfabeto para 4 bits



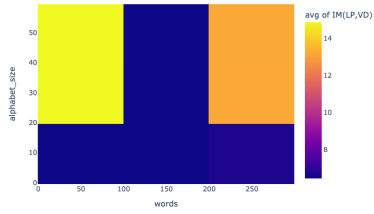
IM en función del tamaño de palabra y del alfabeto para 5 bits



IM en función del tamaño de palabra y del alfabeto para 6 bits



IM en función del tamaño de palabra y del alfabeto para 7 bits



IM en función del tamaño de palabra y del alfabeto para 8 bits

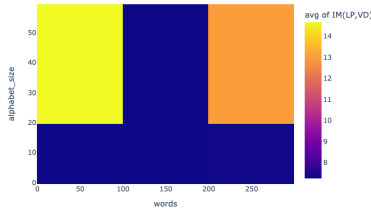


Figura 5.6: Mapa de color de la información mutua variando en función del número de bits, tamaño de palabra w' y tamaño de alfabeto a .

Además el problema de utilizar percentiles equidistantes es que se pierde la posibilidad de que con un espacio menor de combinaciones posible entre todos los símbolos del alfabeto se pueda codificar evento de cierto tipo, como por ejemplo los *spikes* en nuestro algoritmo.

Esto es generalizar y abstraer el problema.

De esta manera Se podían plantear una generalización de lo que nos

5.10 Notas

- Es muy similar a lo que nosotros hemos hecho, solo que este no discretiza en 0 y 1 solo (a no ser que se indique el alfabeto así) y se usa un percentil homogéneo -> es mejor lo que hemos hecho nosotros para detectar los outliers.

6 Summary

In summary, this book has no content whatsoever.

References

Lin, Jessica, Eamonn Keogh, Stefano Lonardi, y Bill Chiu. 2003. «A Symbolic Representation of Time Series, with Implications for Streaming Algorithms». En *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2-11. DMKD '03. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/882082.882086>.

7 Apéndice

7.1 Resultado experimentos para la determinación de los umbrales

7.1.1 Calculando umbrales de ./DatosSinapsisArtificial/TrozoC.zip para la neurona LP

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.281	1.281	-0.105	0.105	130506
1.281	1.645	-0.105	0.135	85508
1.281	1.4	-0.105	0.115	114469
1.281	1.956	-0.105	0.161	61887
1.281	2.57	-0.105	0.211	43399
1.281	3.891	-0.105	0.320	31815
1.281	4.892	-0.105	0.402	31065
1.281	5.326	-0.105	0.437	30766
1.645	1.281	-0.135	0.105	101376
1.645	1.645	-0.135	0.135	76434
1.645	1.4	-0.135	0.115	93535
1.645	1.956	-0.135	0.161	59420
1.645	2.57	-0.135	0.211	43121
1.645	3.891	-0.135	0.320	31814
1.645	4.892	-0.135	0.402	31065
1.645	5.326	-0.135	0.437	30766
1.4	1.281	-0.115	0.105	121840
1.4	1.645	-0.115	0.135	83479
1.4	1.4	-0.115	0.115	108729
1.4	1.956	-0.115	0.161	61457
1.4	2.57	-0.115	0.211	43370
1.4	3.891	-0.115	0.320	31814
1.4	4.892	-0.115	0.402	31065
1.4	5.326	-0.115	0.437	30766
1.956	1.281	-0.161	0.105	81617
1.956	1.645	-0.161	0.135	65504

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.956	1.4	-0.161	0.115	76494
1.956	1.956	-0.161	0.161	54464
1.956	2.57	-0.161	0.211	41905
1.956	3.891	-0.161	0.320	31813
1.956	4.892	-0.161	0.402	31065
1.956	5.326	-0.161	0.437	30766
2.57	1.281	-0.211	0.105	63866
2.57	1.645	-0.211	0.135	51840
2.57	1.4	-0.211	0.115	59820
2.57	1.956	-0.211	0.161	45446
2.57	2.57	-0.211	0.211	38986
2.57	3.891	-0.211	0.320	31799
2.57	4.892	-0.211	0.402	31064
2.57	5.326	-0.211	0.437	30765
3.891	1.281	-0.320	0.105	35110
3.891	1.645	-0.320	0.135	33516
3.891	1.4	-0.320	0.115	34716
3.891	1.956	-0.320	0.161	32306
3.891	2.57	-0.320	0.211	31310
3.891	3.891	-0.320	0.320	30308
3.891	4.892	-0.320	0.402	30135
3.891	5.326	-0.320	0.437	29920
4.892	1.281	-0.402	0.105	10539
4.892	1.645	-0.402	0.135	10498
4.892	1.4	-0.402	0.115	10536
4.892	1.956	-0.402	0.161	10408
4.892	2.57	-0.402	0.211	10343
4.892	3.891	-0.402	0.320	10252
4.892	4.892	-0.402	0.402	10241
4.892	5.326	-0.402	0.437	10232
5.326	1.281	-0.437	0.105	3386
5.326	1.645	-0.437	0.135	3386
5.326	1.4	-0.437	0.115	3386
5.326	1.956	-0.437	0.161	3379
5.326	2.57	-0.437	0.211	3375
5.326	3.891	-0.437	0.320	3359
5.326	4.892	-0.437	0.402	3356
5.326	5.326	-0.437	0.437	3354

7.1.2 Calculando umbrales de ./DatosSinapsisArtificial/TrozoC.zip para la neurona VD

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.281	1.281	0.002	0.086	44982
1.281	1.645	0.002	0.098	28608
1.281	1.4	0.002	0.090	38496
1.281	1.956	0.002	0.108	23721
1.281	2.57	0.002	0.128	22551
1.281	3.891	0.002	0.172	22376
1.281	4.892	0.002	0.205	22325
1.281	5.326	0.002	0.219	22291
1.645	1.281	-0.010	0.086	34485
1.645	1.645	-0.010	0.098	26859
1.645	1.4	-0.010	0.090	31383
1.645	1.956	-0.010	0.108	23631
1.645	2.57	-0.010	0.128	22540
1.645	3.891	-0.010	0.172	22375
1.645	4.892	-0.010	0.205	22325
1.645	5.326	-0.010	0.219	22291
1.4	1.281	-0.002	0.086	39690
1.4	1.645	-0.002	0.098	28001
1.4	1.4	-0.002	0.090	35282
1.4	1.956	-0.002	0.108	23698
1.4	2.57	-0.002	0.128	22548
1.4	3.891	-0.002	0.172	22376
1.4	4.892	-0.002	0.205	22325
1.4	5.326	-0.002	0.219	22291
1.956	1.281	-0.020	0.086	33037
1.956	1.645	-0.020	0.098	26270
1.956	1.4	-0.020	0.090	30153
1.956	1.956	-0.020	0.108	23532
1.956	2.57	-0.020	0.128	22529
1.956	3.891	-0.020	0.172	22375
1.956	4.892	-0.020	0.205	22325
1.956	5.326	-0.020	0.219	22291
2.57	1.281	-0.041	0.086	32649
2.57	1.645	-0.041	0.098	26007
2.57	1.4	-0.041	0.090	29796
2.57	1.956	-0.041	0.108	23391
2.57	2.57	-0.041	0.128	22484

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
2.57	3.891	-0.041	0.172	22357
2.57	4.892	-0.041	0.205	22318
2.57	5.326	-0.041	0.219	22288
3.891	1.281	-0.084	0.086	28061
3.891	1.645	-0.084	0.098	23717
3.891	1.4	-0.084	0.090	26241
3.891	1.956	-0.084	0.108	21945
3.891	2.57	-0.084	0.128	21324
3.891	3.891	-0.084	0.172	21265
3.891	4.892	-0.084	0.205	21246
3.891	5.326	-0.084	0.219	21244
4.892	1.281	-0.117	0.086	12175
4.892	1.645	-0.117	0.098	11921
4.892	1.4	-0.117	0.090	12069
4.892	1.956	-0.117	0.108	11801
4.892	2.57	-0.117	0.128	11748
4.892	3.891	-0.117	0.172	11728
4.892	4.892	-0.117	0.205	11719
4.892	5.326	-0.117	0.219	11719
5.326	1.281	-0.131	0.086	5673
5.326	1.645	-0.131	0.098	5647
5.326	1.4	-0.131	0.090	5668
5.326	1.956	-0.131	0.108	5637
5.326	2.57	-0.131	0.128	5632
5.326	3.891	-0.131	0.172	5621
5.326	4.892	-0.131	0.205	5613
5.326	5.326	-0.131	0.219	5613

7.1.3 Calculando umbrales de ./DatosSinapsisArtificial/TrozoR.zip para la neurona LP

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.281	1.281	-0.283	0.216	97572
1.281	1.645	-0.283	0.287	70454
1.281	1.4	-0.283	0.239	89176
1.281	1.956	-0.283	0.348	53151
1.281	2.57	-0.283	0.467	33900
1.281	3.891	-0.283	0.725	25571

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.281	4.892	-0.283	0.920	24248
1.281	5.326	-0.283	1.004	22390
1.645	1.281	-0.354	0.216	73593
1.645	1.645	-0.354	0.287	58852
1.645	1.4	-0.354	0.239	69000
1.645	1.956	-0.354	0.348	48757
1.645	2.57	-0.354	0.467	33684
1.645	3.891	-0.354	0.725	25569
1.645	4.892	-0.354	0.920	24248
1.645	5.326	-0.354	1.004	22390
1.4	1.281	-0.306	0.216	90145
1.4	1.645	-0.306	0.287	67590
1.4	1.4	-0.306	0.239	83219
1.4	1.956	-0.306	0.348	52341
1.4	2.57	-0.306	0.467	33864
1.4	3.891	-0.306	0.725	25571
1.4	4.892	-0.306	0.920	24248
1.4	5.326	-0.306	1.004	22390
1.956	1.281	-0.415	0.216	58975
1.956	1.645	-0.415	0.287	48397
1.956	1.4	-0.415	0.239	55450
1.956	1.956	-0.415	0.348	42416
1.956	2.57	-0.415	0.467	32857
1.956	3.891	-0.415	0.725	25547
1.956	4.892	-0.415	0.920	24239
1.956	5.326	-0.415	1.004	22389
2.57	1.281	-0.534	0.216	45236
2.57	1.645	-0.534	0.287	36911
2.57	1.4	-0.534	0.239	42264
2.57	1.956	-0.534	0.348	33329
2.57	2.57	-0.534	0.467	29663
2.57	3.891	-0.534	0.725	25258
2.57	4.892	-0.534	0.920	24076
2.57	5.326	-0.534	1.004	22299
3.891	1.281	-0.792	0.216	3026
3.891	1.645	-0.792	0.287	3011
3.891	1.4	-0.792	0.239	3025
3.891	1.956	-0.792	0.348	2993
3.891	2.57	-0.792	0.467	2963
3.891	3.891	-0.792	0.725	2937

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
3.891	4.892	-0.792	0.920	2927
3.891	5.326	-0.792	1.004	2909
4.892	1.281	-0.987	0.216	167
4.892	1.645	-0.987	0.287	167
4.892	1.4	-0.987	0.239	167
4.892	1.956	-0.987	0.348	167
4.892	2.57	-0.987	0.467	166
4.892	3.891	-0.987	0.725	166
4.892	4.892	-0.987	0.920	166
4.892	5.326	-0.987	1.004	166
5.326	1.281	-1.071	0.216	46
5.326	1.645	-1.071	0.287	46
5.326	1.4	-1.071	0.239	46
5.326	1.956	-1.071	0.348	46
5.326	2.57	-1.071	0.467	46
5.326	3.891	-1.071	0.725	46
5.326	4.892	-1.071	0.920	46
5.326	5.326	-1.071	1.004	46

7.1.4 Calculando umbrales de ./DatosSinapsisArtificial/TrozoR.zip para la neurona VD

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.281	1.281	0.001	0.086	37189
1.281	1.645	0.001	0.098	23641
1.281	1.4	0.001	0.090	31884
1.281	1.956	0.001	0.109	19765
1.281	2.57	0.001	0.129	18854
1.281	3.891	0.001	0.173	18705
1.281	4.892	0.001	0.206	18663
1.281	5.326	0.001	0.220	18632
1.645	1.281	-0.011	0.086	28713
1.645	1.645	-0.011	0.098	22327
1.645	1.4	-0.011	0.090	26161
1.645	1.956	-0.011	0.109	19693
1.645	2.57	-0.011	0.129	18846
1.645	3.891	-0.011	0.173	18704
1.645	4.892	-0.011	0.206	18663

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.645	5.326	-0.011	0.220	18632
1.4	1.281	-0.002	0.086	32905
1.4	1.645	-0.002	0.098	23180
1.4	1.4	-0.002	0.090	29306
1.4	1.956	-0.002	0.109	19744
1.4	2.57	-0.002	0.129	18852
1.4	3.891	-0.002	0.173	18705
1.4	4.892	-0.002	0.206	18663
1.4	5.326	-0.002	0.220	18632
1.956	1.281	-0.021	0.086	27512
1.956	1.645	-0.021	0.098	21846
1.956	1.4	-0.021	0.090	25143
1.956	1.956	-0.021	0.109	19615
1.956	2.57	-0.021	0.129	18838
1.956	3.891	-0.021	0.173	18704
1.956	4.892	-0.021	0.206	18663
1.956	5.326	-0.021	0.220	18632
2.57	1.281	-0.041	0.086	27159
2.57	1.645	-0.041	0.098	21603
2.57	1.4	-0.041	0.090	24817
2.57	1.956	-0.041	0.109	19482
2.57	2.57	-0.041	0.129	18799
2.57	3.891	-0.041	0.173	18688
2.57	4.892	-0.041	0.206	18654
2.57	5.326	-0.041	0.220	18626
3.891	1.281	-0.085	0.086	23076
3.891	1.645	-0.085	0.098	19535
3.891	1.4	-0.085	0.090	21615
3.891	1.956	-0.085	0.109	18139
3.891	2.57	-0.085	0.129	17686
3.891	3.891	-0.085	0.173	17636
3.891	4.892	-0.085	0.206	17618
3.891	5.326	-0.085	0.220	17616
4.892	1.281	-0.118	0.086	9482
4.892	1.645	-0.118	0.098	9284
4.892	1.4	-0.118	0.090	9401
4.892	1.956	-0.118	0.109	9201
4.892	2.57	-0.118	0.129	9167
4.892	3.891	-0.118	0.173	9148
4.892	4.892	-0.118	0.206	9139

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
4.892	5.326	-0.118	0.220	9139
5.326	1.281	-0.133	0.086	4443
5.326	1.645	-0.133	0.098	4420
5.326	1.4	-0.133	0.090	4438
5.326	1.956	-0.133	0.109	4411
5.326	2.57	-0.133	0.129	4406
5.326	3.891	-0.133	0.173	4396
5.326	4.892	-0.133	0.206	4389
5.326	5.326	-0.133	0.220	4389

7.1.5 Calculando umbrales de ./DatosSinapsisArtificial/TrozoG.zip para la neurona LP

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.281	1.281	-0.104	0.104	104888
1.281	1.645	-0.104	0.133	68612
1.281	1.4	-0.104	0.114	92078
1.281	1.956	-0.104	0.159	51140
1.281	2.57	-0.104	0.208	36986
1.281	3.891	-0.104	0.316	27454
1.281	4.892	-0.104	0.397	26457
1.281	5.326	-0.104	0.432	26051
1.645	1.281	-0.133	0.104	81156
1.645	1.645	-0.133	0.133	60412
1.645	1.4	-0.133	0.114	74105
1.645	1.956	-0.133	0.159	48783
1.645	2.57	-0.133	0.208	36803
1.645	3.891	-0.133	0.316	27453
1.645	4.892	-0.133	0.397	26457
1.645	5.326	-0.133	0.432	26051
1.4	1.281	-0.114	0.104	96386
1.4	1.645	-0.114	0.133	66324
1.4	1.4	-0.114	0.114	86071
1.4	1.956	-0.114	0.159	50638
1.4	2.57	-0.114	0.208	36965
1.4	3.891	-0.114	0.316	27454
1.4	4.892	-0.114	0.397	26457
1.4	5.326	-0.114	0.432	26051

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.956	1.281	-0.159	0.104	66653
1.956	1.645	-0.159	0.133	51944
1.956	1.4	-0.159	0.114	61382
1.956	1.956	-0.159	0.159	44720
1.956	2.57	-0.159	0.208	35915
1.956	3.891	-0.159	0.316	27453
1.956	4.892	-0.159	0.397	26457
1.956	5.326	-0.159	0.432	26051
2.57	1.281	-0.208	0.104	53512
2.57	1.645	-0.208	0.133	41501
2.57	1.4	-0.208	0.114	48946
2.57	1.956	-0.208	0.159	37295
2.57	2.57	-0.208	0.208	32872
2.57	3.891	-0.208	0.316	27425
2.57	4.892	-0.208	0.397	26456
2.57	5.326	-0.208	0.432	26050
3.891	1.281	-0.316	0.104	40285
3.891	1.645	-0.316	0.133	33357
3.891	1.4	-0.316	0.114	37792
3.891	1.956	-0.316	0.159	31056
3.891	2.57	-0.316	0.208	28910
3.891	3.891	-0.316	0.316	26516
3.891	4.892	-0.316	0.397	25889
3.891	5.326	-0.316	0.432	25547
4.892	1.281	-0.397	0.104	4852
4.892	1.645	-0.397	0.133	4824
4.892	1.4	-0.397	0.114	4849
4.892	1.956	-0.397	0.159	4777
4.892	2.57	-0.397	0.208	4726
4.892	3.891	-0.397	0.316	4676
4.892	4.892	-0.397	0.397	4659
4.892	5.326	-0.397	0.432	4649
5.326	1.281	-0.432	0.104	2055
5.326	1.645	-0.432	0.133	2053
5.326	1.4	-0.432	0.114	2055
5.326	1.956	-0.432	0.159	2042
5.326	2.57	-0.432	0.208	2034
5.326	3.891	-0.432	0.316	2021
5.326	4.892	-0.432	0.397	2021
5.326	5.326	-0.432	0.432	2021

7.1.6 Calculando umbrales de ./DatosSinapsisArtificial/TrozoG.zip para la neurona VD

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
1.281	1.281	0.001	0.086	28195
1.281	1.645	0.001	0.098	18800
1.281	1.4	0.001	0.090	24436
1.281	1.956	0.001	0.109	16283
1.281	2.57	0.001	0.129	15689
1.281	3.891	0.001	0.173	15662
1.281	4.892	0.001	0.207	15661
1.281	5.326	0.001	0.221	15660
1.645	1.281	-0.012	0.086	23705
1.645	1.645	-0.012	0.098	18253
1.645	1.4	-0.012	0.090	21474
1.645	1.956	-0.012	0.109	16251
1.645	2.57	-0.012	0.129	15689
1.645	3.891	-0.012	0.173	15662
1.645	4.892	-0.012	0.207	15661
1.645	5.326	-0.012	0.221	15660
1.4	1.281	-0.003	0.086	25590
1.4	1.645	-0.003	0.098	18574
1.4	1.4	-0.003	0.090	22878
1.4	1.956	-0.003	0.109	16269
1.4	2.57	-0.003	0.129	15689
1.4	3.891	-0.003	0.173	15662
1.4	4.892	-0.003	0.207	15661
1.4	5.326	-0.003	0.221	15660
1.956	1.281	-0.022	0.086	23312
1.956	1.645	-0.022	0.098	18137
1.956	1.4	-0.022	0.090	21163
1.956	1.956	-0.022	0.109	16230
1.956	2.57	-0.022	0.129	15689
1.956	3.891	-0.022	0.173	15662
1.956	4.892	-0.022	0.207	15661
1.956	5.326	-0.022	0.221	15660
2.57	1.281	-0.042	0.086	23229
2.57	1.645	-0.042	0.098	18107
2.57	1.4	-0.042	0.090	21089
2.57	1.956	-0.042	0.109	16226
2.57	2.57	-0.042	0.129	15689

Distancia del umbral bajo	Distancia alta	Umbral bajo	Umbral alto	Número de <i>spikes</i>
2.57	3.891	-0.042	0.173	15662
2.57	4.892	-0.042	0.207	15661
2.57	5.326	-0.042	0.221	15660
3.891	1.281	-0.087	0.086	22593
3.891	1.645	-0.087	0.098	17877
3.891	1.4	-0.087	0.090	20637
3.891	1.956	-0.087	0.109	16102
3.891	2.57	-0.087	0.129	15598
3.891	3.891	-0.087	0.173	15571
3.891	4.892	-0.087	0.207	15571
3.891	5.326	-0.087	0.221	15571
4.892	1.281	-0.120	0.086	15080
4.892	1.645	-0.120	0.098	13814
4.892	1.4	-0.120	0.090	14584
4.892	1.956	-0.120	0.109	13283
4.892	2.57	-0.120	0.129	13137
4.892	3.891	-0.120	0.173	13127
4.892	4.892	-0.120	0.207	13127
4.892	5.326	-0.120	0.221	13127
5.326	1.281	-0.134	0.086	9828
5.326	1.645	-0.134	0.098	9566
5.326	1.4	-0.134	0.090	9730
5.326	1.956	-0.134	0.109	9459
5.326	2.57	-0.134	0.129	9427
5.326	3.891	-0.134	0.173	9424
5.326	4.892	-0.134	0.207	9424
5.326	5.326	-0.134	0.221	9424

7.2 Información mutua obtenida para la codificación SAX

7.2.1 Trozo A

trozo	bits	words	alphabet size	IM(LP,VD)
C	1	11	1	0.00024815504515718345
C	2	11	1	0.001028528475531476
C	3	11	1	0.0021732004578957076
C	4	11	1	0.003579663704288194
C	5	11	1	0.005386666234739712

trozo	bits	words	alphabet size	IM(LP,VD)
C	6	11	1	0.007695825616436114
C	7	11	1	0.010386906300554832
C	8	11	1	0.013729273621681859
C	1	87	1	0.0002849437766737073
C	2	87	1	0.0010789203951092619
C	3	87	1	0.0024516116162923707
C	4	87	1	0.004059343543587723
C	5	87	1	0.006087685587868652
C	6	87	1	0.00940811921254836
C	7	87	1	0.013262453561221355
C	8	87	1	0.01610577817892711
C	1	187	1	0.00010869489163545243
C	2	187	1	0.0007406252047386097
C	3	187	1	0.001830320692957077
C	4	187	1	0.0036377228714716825
C	5	187	1	0.006689120569749907
C	6	187	1	0.010156060947295709
C	7	187	1	0.01563190742403764
C	8	187	1	0.020783911704893132
C	1	11	2	0.0005182298593626733
C	2	11	2	0.0024795852806711594
C	3	11	2	0.0067956803236759455
C	4	11	2	0.013927773912149277
C	5	11	2	0.03151171221116478
C	6	11	2	0.08216009739659569
C	7	11	2	0.22058277844140228
C	8	11	2	0.5569383137386641
C	1	87	2	0.00035897039574805945
C	2	87	2	0.001361959194343676
C	3	87	2	0.006346304795460611
C	4	87	2	0.02673803655749296
C	5	87	2	0.1228606234356242
C	6	87	2	0.41818007907327726
C	7	87	2	1.078962133677889
C	8	87	2	2.2358763601826492
C	1	187	2	0.00021414267703434575
C	2	187	2	0.0018672074628094393
C	3	187	2	0.011107682310548483
C	4	187	2	0.05337350049381939
C	5	187	2	0.21877495621872
C	6	187	2	0.6321478810593355

trozo	bits	words	alphabet size	IM(LP,VD)
C	7	187	2	1.455086240946569
C	8	187	2	2.755969934814372
C	1	11	5	0.002644232630698795
C	2	11	5	0.009773159338747561
C	3	11	5	0.044856554473932775
C	4	11	5	0.2897626915538929
C	5	11	5	1.1166215209535455
C	6	11	5	2.737896224607141
C	7	11	5	4.885394705950354
C	8	11	5	7.1549160545849375
C	1	87	5	0.002675269056581442
C	2	87	5	0.016125860153871407
C	3	87	5	0.19732480171007438
C	4	87	5	1.163783987589479
C	5	87	5	3.2532443838401033
C	6	87	5	5.901220205850953
C	7	87	5	8.290037828698518
C	8	87	5	10.059693429503145
C	1	187	5	0.002117483486257399
C	2	187	5	0.021992328126568594
C	3	187	5	0.4190740942927267
C	4	187	5	2.3199764359542314
C	5	187	5	5.473442410107175
C	6	187	5	8.48113421257955
C	7	187	5	10.660943808648625
C	8	187	5	12.05381465773515
C	1	11	10	0.004024574980462603
C	2	11	10	0.023398301841709213
C	3	11	10	0.4801280544736759
C	4	11	10	2.745372664491679
C	5	11	10	6.376897634191838
C	6	11	10	9.953007527809302
C	7	11	10	12.711268637131639
C	8	11	10	14.541391428139981
C	1	87	10	0.003951176148957458
C	2	87	10	0.09534519380188833
C	3	87	10	2.0148750120601076
C	4	87	10	6.894191250706406
C	5	87	10	11.278625052808277
C	6	87	10	13.65511399742252
C	7	87	10	14.490412059901923

trozo	bits	words	alphabet size	IM(LP,VD)
C	8	87	10	14.628726620737421
C	1	187	10	0.0035871669646834192
C	2	187	10	0.17058251997290164
C	3	187	10	2.70463717000497
C	4	187	10	7.562490477796784
C	5	187	10	11.296209954929086
C	6	187	10	13.083683889229825
C	7	187	10	13.578574811874777
C	8	187	10	13.613191381304558
C	1	11	20	0.005257504383962441
C	2	11	20	0.1328026682875656
C	3	11	20	2.645469769828882
C	4	11	20	7.637184867247232
C	5	11	20	11.9503436289455
C	6	11	20	14.566524678120338
C	7	11	20	15.786440743543693
C	8	11	20	16.268007904068543
C	1	87	20	0.006435332990818665
C	2	87	20	0.6959842625990547
C	3	87	20	6.3716750385889185
C	4	87	20	12.237750569583373
C	5	87	20	14.673394682137074
C	6	87	20	15.05844560118894
C	7	87	20	14.961597675367718
C	8	87	20	14.793223333549019
C	1	187	20	0.006898247478708264
C	2	187	20	1.0842701127646777
C	3	187	20	7.215229873872396
C	4	187	20	12.255891045618043
C	5	187	20	13.877530616433193
C	6	187	20	14.034483955093329
C	7	187	20	13.875688330654611
C	8	187	20	13.693293308041325
C	1	11	50	0.006208055597015871
C	2	11	50	1.2384320571645908
C	3	11	50	8.463614924696365
C	4	11	50	14.628385841768335
C	5	11	50	16.86687773936527
C	6	11	50	17.34927554153607
C	7	11	50	17.464702200138227
C	8	11	50	17.477040225545803

trozo	bits	words	alphabet size	IM(LP,VD)
C	1	87	50	0.014035859612665291
C	2	87	50	3.3208460073086385
C	3	87	50	11.68470901812871
C	4	87	50	15.093791178005088
C	5	87	50	15.349300274054544
C	6	87	50	15.183218447931885
C	7	87	50	14.98489727147203
C	8	87	50	14.79800186709187
C	1	187	50	0.022431582741583966
C	2	187	50	4.63454577701857
C	3	187	50	12.63017991862702
C	4	187	50	14.500063268424503
C	5	187	50	14.360809627001219
C	6	187	50	14.109886483299473
C	7	187	50	13.88817244967031
C	8	187	50	13.695554557758664

7.2.2 Trozo R

trozo	bits	words	alphabet size	IM(LP,VD)
R	1	17	1	0.00024012310475840515
R	2	17	1	0.0009182109736400879
R	3	17	1	0.002007256336612806
R	4	17	1	0.0034306641732579912
R	5	17	1	0.005187612417107967
R	6	17	1	0.007696391363570809
R	7	17	1	0.010384489594184032
R	8	17	1	0.01343699675738641
R	1	42	1	0.0001543889357999706
R	2	42	1	0.0006346228647774455
R	3	42	1	0.0016006647501978222
R	4	42	1	0.0023673147805025607
R	5	42	1	0.003890434575817059
R	6	42	1	0.005802578365191158
R	7	42	1	0.007824930487269732
R	8	42	1	0.010252248398820107
R	1	125	1	0.00010938470868754324
R	2	125	1	0.0003318369190691284
R	3	125	1	0.0008782367128054025
R	4	125	1	0.0018704257551911407

trozo	bits	words	alphabet size	IM(LP,VD)
R	5	125	1	0.0040093427516307845
R	6	125	1	0.0070559861152088565
R	7	125	1	0.012166963830938737
R	8	125	1	0.019087690622356046
R	1	17	2	0.0006400023165884505
R	2	17	2	0.003620383241629277
R	3	17	2	0.009712442567233914
R	4	17	2	0.020252289012088198
R	5	17	2	0.047632679216361495
R	6	17	2	0.1282657203426787
R	7	17	2	0.3601899893188545
R	8	17	2	0.8796357566388693
R	1	42	2	0.00035101698339090603
R	2	42	2	0.0017426427337898787
R	3	42	2	0.00580378336666687
R	4	42	2	0.020413492266053268
R	5	42	2	0.08544084190304879
R	6	42	2	0.3044749601621888
R	7	42	2	0.8382897718827156
R	8	42	2	1.8259444710530524
R	1	125	2	0.00039111545913561585
R	2	125	2	0.0017642335933416575
R	3	125	2	0.007372844606811313
R	4	125	2	0.039136921498331034
R	5	125	2	0.15747545214627579
R	6	125	2	0.5055822386874365
R	7	125	2	1.2686344962945615
R	8	125	2	2.548445082974867
R	1	17	5	0.0038005342815585763
R	2	17	5	0.011831964141347129
R	3	17	5	0.05686177435162598
R	4	17	5	0.35478298852674506
R	5	17	5	1.378979687284966
R	6	17	5	3.3371993268936464
R	7	17	5	5.802233907563089
R	8	17	5	8.194009525106772
R	1	42	5	0.0029994503726804567
R	2	42	5	0.013292488264040614
R	3	42	5	0.14636574027242588
R	4	42	5	1.0865046360864064
R	5	42	5	3.4628763555603985

trozo	bits	words	alphabet size	IM(LP,VD)
R	6	42	5	6.613873636987028
R	7	42	5	9.412977224024232
R	8	42	5	11.438334440244244
R	1	125	5	0.002839834548706399
R	2	125	5	0.02013958271322558
R	3	125	5	0.29264903115308094
R	4	125	5	1.546954703008426
R	5	125	5	3.9296082639979293
R	6	125	5	6.596990690860762
R	7	125	5	8.798106896394223
R	8	125	5	10.338395335812393
R	1	17	10	0.005404722325938138
R	2	17	10	0.03367489929994427
R	3	17	10	0.5601996232551638
R	4	17	10	3.072853416780088
R	5	17	10	7.024016280270072
R	6	17	10	10.738811755054666
R	7	17	10	13.435677923932666
R	8	17	10	15.051816561548648
R	1	42	10	0.004340888061840964
R	2	42	10	0.06560099282434884
R	3	42	10	1.6552401840046826
R	4	42	10	6.180364106822761
R	5	42	10	10.65878394219936
R	6	42	10	13.390077857041684
R	7	42	10	14.651440715462355
R	8	42	10	15.133808380961383
R	1	125	10	0.003874083010751761
R	2	125	10	0.13988406249165664
R	3	125	10	2.168132764780463
R	4	125	10	6.666235349179114
R	5	125	10	10.455805239597122
R	6	125	10	12.485483628017766
R	7	125	10	13.412291163554418
R	8	125	10	13.71764410455522
R	1	17	20	0.006404399097171165
R	2	17	20	0.1889768704904462
R	3	17	20	3.1618889774588084
R	4	17	20	8.643712889329741
R	5	17	20	13.085164353960923
R	6	17	20	15.347639925999754

trozo	bits	words	alphabet size	IM(LP,VD)
R	7	17	20	16.061772234700825
R	8	17	20	16.218409500374893
R	1	42	20	0.00581938282667771
R	2	42	20	0.4964436711508764
R	3	42	20	5.796116497621668
R	4	42	20	11.870762115295037
R	5	42	20	14.881339948384294
R	6	42	20	15.598776808514229
R	7	42	20	15.612412794783639
R	8	42	20	15.502510287323666
R	1	125	20	0.006091756415940175
R	2	125	20	1.017698170758015
R	3	125	20	7.113961248222115
R	4	125	20	12.28640261199174
R	5	125	20	13.951908447466142
R	6	125	20	14.13933925774478
R	7	125	20	14.062384747858633
R	8	125	20	13.941960915478596
R	1	17	50	0.009395435061712476
R	2	17	50	1.6138756675382062
R	3	17	50	9.373878786731478
R	4	17	50	15.176425920846842
R	5	17	50	16.79036583468885
R	6	17	50	16.977233844964285
R	7	17	50	16.923350480827118
R	8	17	50	16.813161207140322
R	1	42	50	0.010517169466353948
R	2	42	50	2.890153388869564
R	3	42	50	11.447447926865344
R	4	42	50	15.47416373448316
R	5	42	50	16.053265169984382
R	6	42	50	15.94096814739467
R	7	42	50	15.754502858632579
R	8	42	50	15.570506218470847
R	1	125	50	0.018673454891793284
R	2	125	50	4.136439038062516
R	3	125	50	12.051220887351379
R	4	125	50	14.412317681476289
R	5	125	50	14.498659841194835
R	6	125	50	14.345742950993419
R	7	125	50	14.162681350456022

trozo	bits	words	alphabet size	IM(LP,VD)
R	8	125	50	13.990863779265805

7.2.3 Trozo G

trozo	bits	words	alphabet size	IM(LP,VD)
G	1	66	1	0.00015869819397110185
G	2	66	1	0.0008295083835761496
G	3	66	1	0.0017875778380064267
G	4	66	1	0.0030830623799174006
G	5	66	1	0.005305752877250214
G	6	66	1	0.007451544001771948
G	7	66	1	0.010680290570000528
G	8	66	1	0.013829155451792863
G	1	93	1	0.00036552969034847616
G	2	93	1	0.0010641448823445199
G	3	93	1	0.0027101225889893943
G	4	93	1	0.004600974009801195
G	5	93	1	0.006709568901130947
G	6	93	1	0.010420501444593278
G	7	93	1	0.015036655711274438
G	8	93	1	0.020462283788194924
G	1	248	1	0.0002325560137608207
G	2	248	1	0.0008151765727507643
G	3	248	1	0.0018088033979524187
G	4	248	1	0.0043167179318872595
G	5	248	1	0.008204682111097261
G	6	248	1	0.012267204759849104
G	7	248	1	0.0194253829985076
G	8	248	1	0.029502747339232194
G	1	66	2	0.0003841994900461998
G	2	66	2	0.001968680745290996
G	3	66	2	0.006895588780301942
G	4	66	2	0.02830509548175897
G	5	66	2	0.11427353552211628
G	6	66	2	0.37659830886420664
G	7	66	2	0.9430328752799486
G	8	66	2	1.9322229125606967
G	1	93	2	0.00035877026995212447
G	2	93	2	0.0016510620857088654
G	3	93	2	0.006940785387367754

trozo	bits	words	alphabet size	IM(LP,VD)
G	4	93	2	0.029936969987602424
G	5	93	2	0.14114168290199913
G	6	93	2	0.4625762394976949
G	7	93	2	1.1425493510136064
G	8	93	2	2.3389238562929897
G	1	248	2	0.0002529999741076594
G	2	248	2	0.0019012131807105703
G	3	248	2	0.013695232826538728
G	4	248	2	0.07687985286959531
G	5	248	2	0.307054120580478
G	6	248	2	0.897543778789224
G	7	248	2	2.062395249751786
G	8	248	2	3.697229783545259
G	1	66	5	0.00179169143094704
G	2	66	5	0.012082762994605645
G	3	66	5	0.17389670192300244
G	4	66	5	1.3318479844673163
G	5	66	5	4.158138791013769
G	6	66	5	7.5258040771209185
G	7	66	5	10.42292484650601
G	8	66	5	12.420186697705217
G	1	93	5	0.0015215832726322986
G	2	93	5	0.014492473733714206
G	3	93	5	0.2597471099345956
G	4	93	5	1.7717605019456268
G	5	93	5	4.860269325836931
G	6	93	5	8.211673543613077
G	7	93	5	10.833191453250222
G	8	93	5	12.470627466244986
G	1	248	5	0.0014586773772640171
G	2	248	5	0.029868605736570686
G	3	248	5	0.5860052593998812
G	4	248	5	3.184915502592448
G	5	248	5	6.914191794372126
G	6	248	5	10.002779936464583
G	7	248	5	11.822029571831516
G	8	248	5	12.548229784721503
G	1	66	10	0.0035786839044034124
G	2	66	10	0.08460242289471509
G	3	66	10	1.80307331543586
G	4	66	10	6.445952829493091

trozo	bits	words	alphabet size	IM(LP,VD)
G	5	66	10	10.888901549815175
G	6	66	10	13.529780346698837
G	7	66	10	14.506668707097429
G	8	66	10	14.691858597352342
G	1	93	10	0.0025979164915019837
G	2	93	10	0.10372699106430083
G	3	93	10	1.7751611844906225
G	4	93	10	6.007876898902136
G	5	93	10	10.112216015017303
G	6	93	10	12.599022602184968
G	7	93	10	13.751762216240092
G	8	93	10	14.148303585034547
G	1	248	10	0.0031999120538808157
G	2	248	10	0.23019959175370452
G	3	248	10	2.9583217458783544
G	4	248	10	7.775434718404904
G	5	248	10	11.217138667219219
G	6	248	10	12.666265878970885
G	7	248	10	13.011372874475315
G	8	248	10	12.971670082320308
G	1	66	20	0.005124931252082021
G	2	66	20	0.6268700860983252
G	3	66	20	6.142163616636093
G	4	66	20	12.162779436022925
G	5	66	20	14.80569517452694
G	6	66	20	15.18058620616491
G	7	66	20	15.077938236441621
G	8	66	20	14.911714854813889
G	1	93	20	0.004412120353683768
G	2	93	20	0.7743603531339005
G	3	93	20	6.505231107559526
G	4	93	20	12.116640712087463
G	5	93	20	14.365656765552881
G	6	93	20	14.730063185787497
G	7	93	20	14.603832058231436
G	8	93	20	14.424503449320325
G	1	248	20	0.0061588624721053975
G	2	248	20	1.5208236327682503
G	3	248	20	8.30840510097624
G	4	248	20	12.715696841784856
G	5	248	20	13.554978382423524

trozo	bits	words	alphabet size	IM(LP,VD)
G	6	248	20	13.411675824379005
G	7	248	20	13.202840997238576
G	8	248	20	13.0113345137833
G	1	66	50	0.011551791009663859
G	2	66	50	3.055780814245491
G	3	66	50	11.255708040195657
G	4	66	50	14.905466984414861
G	5	66	50	15.460422336875213
G	6	66	50	15.318633606349803
G	7	66	50	15.11265215603812
G	8	66	50	14.921053807632985
G	1	93	50	0.013848690209110615
G	2	93	50	3.555942153230273
G	3	93	50	11.780124006572116
G	4	93	50	14.89037375279692
G	5	93	50	15.059765297436794
G	6	93	50	14.838180608798826
G	7	93	50	14.618927755096315
G	8	93	50	14.426592364446785
G	1	248	50	0.030383044492735323
G	2	248	50	4.880680346044221
G	3	248	50	12.287520077380595
G	4	248	50	13.865996487237746
G	5	248	50	13.682446510117469
G	6	248	50	13.42616354236299
G	7	248	50	13.204112587786417
G	8	248	50	13.011576703175404