Estudio de patrones de disparo neuronales

Belén Citoler Berdala

Trabajo de fin de máster en
Matemática Computacional
Universitat Jaume I

Directora del trabajo: Marina Martínez García

Octubre 2020

Resumen

La versatilidad de las matemáticas es la propiedad que hace de ellas una ciencia tan especial y cada vez más presente en nuestra sociedad. Tiene muchas aplicaciones en ámbitos muy diversos, desde la economía hasta la medicina.

En este proyecto se tratará temas relacionados con la neurociencia. Un campo multidisciplinar, en el que intervienen investigadores e investigadoras de muchas ciencias; entre ellas, las matemáticas.

La neurociencia es un campo muy diverso y complejo. Es por ello, que en el presente trabajo, se estudia desde el concepto más básico: las neuronas. Veremos como a través de un algoritmo es posible estudiar su comportamiento para ver qué información codifican en nuestro cerebro.

En los programas informáticos es fundamental conocer el contexto y realizar un correcto enfoque para poder ser aplicado en la realidad. Por eso, se puede encontrar una breve introducción al interesante mundo de la neurociencia en el Capítulo 1. A través de una recopilación de los principales sucesos históricos, se permitirá ver desde los inicios hasta la instauración de las bases de esta ciencia a manos de Santiago Ramón y Cajal.

Posteriormente, se expone nociones sobre la fisiología más básica como son las neuronas, trataremos de explicar su comportamiento y cómo varía según los estímulos que recibe. También estudiaremos cómo son el tipo de tareas que veremos en el resto del proyecto, así como las distintas áreas del cerebro implicadas. De esta manera se pretende dar sentido y contextualizar los datos que vamos a utilizar.

En el segundo Capítulo se pretende justificar el objetivo del proyecto a través de la cuestión del comportamiento categórico de neuronas. Además, se expone un ejemplo concreto y detallado de una tarea de decisión al que se aplica el algoritmo desarrollado en este mismo capítulo, así como la preparación de los datos y los resultados obtenidos. Se ha considerado realizar una versión en dos dimensiones para facilitar la lectura a través de distintos gráficos y que el algoritmo sea más descriptivo.

Los métodos utilizados a lo largo de todo el procedimiento se exponen en el Capítulo 3, tanto los empleados en la preparación de los datos como en el análisis de los resultados. Siendo los más importantes los dos métodos de clasificación: *spherical k-means* y *variable-centroid clustering.*

Finalmente, en el Capítulo 4 se puede encontrar la parte informática del estudio, una implementación en R desarrollada a partir del código en Python. Se especifican los paquetes usados y los programas requeridos indicando las versiones utilizadas. Y una explicación detallada paso a paso del algoritmo.

Índice general

Capítulo 1

Introducción a la neurociencia

La neurociencia es un campo muy amplio, del que se conoce una mínima parte. El desarrollo de esta ciencia es lento debido a su complejidad, pero los avances tecnológicos nos proporcionan cada vez más recursos para su desarrollo. Su objetivo es el estudio del sistema nervioso y conocer cómo funciona el cerebro y cómo afecta a nuestras conductas, pensamientos y emociones.

Se trata de una ciencia multidisciplinar, puesto que requiere de profesionales de diversos ámbitos como son la biología, la química, la informática o las matemáticas. El conjunto de todas estas ciencias nos permite conocer cómo funcionan los organismos, de esta manera podremos entender ciertas enfermedades y tratar de combatirlas.

En los últimos años, ha incrementado el interés y el apoyo a la neurociencia. En 2013 se lanzaron dos grandes proyectos: BRAIN *(Brain Research through Advancing Innovative Neurotechnologies)* impulsado por Barack Obama y Proyecto Cerebro Humano *(HBP - Human Brain Project)* financiado por la Unión Europea. Ambos proyectos destacan la importancia de los modelos matemáticos, los avances informáticos, así como la unión de distintas disciplinas. Los dos proyectos tuvieron su inicio en el siglo *XXI* y todavía hoy en día se consideran de gran importancia.

A lo largo de este trabajo, vamos a tratar con modelos matemáticos que son aplicados a datos tanto reales como sintéticos. Para poder realizar una correcta interpretación es necesario contextualizar, por ello, es conveniente una breve introducción a la neurociencia.

En primer lugar se expone un breve resumen histórico, se continúa con anatomía básica, posteriormente se trata las tasas de disparo (los datos van a ser este tipo de información) y por último, veremos en qué consisten las tareas de decisión y las distintas áreas del cerebro implicadas.

1.1. Contexto histórico

Actualmente se considera la neurociencia como el estudio del sistema nervioso, pero no ha sido siempre así. Remontándonos a la Antigua Grecia, el filósofo Aristóteles apoyaba la idea de que los procesos intelectuales tenían lugar en el corazón y el cerebro era el que se encargaba de enfriar la sangre que el corazón sobrecalentaba [1].

A mediados del siglo XVII, René Descartes defiende la teoría mecanicista y establece la dualidad cuerpo-alma, según la cual el cerebro controla la conducta.

A principios del siglo XIX, surge la *frenología* que, apoyada por Gall y Brodmann, expone que todos los procesos tienen lugar en el cerebro y que estos tienen un área concreta. Además, propone que el desarrollo de una determinada área implica el incremento del volumen de la parte del cerebro correspondiente. Así comienza una visión dinámica del cerebro, en la que observando la forma y el tamaño de éste se podría llegar a estudiar y potenciar ciertas habilidades del individuo.

Más tarde, aparece el *conectivismo*, corriente que defiende que las funciones complejas conllevan varias zonas conectadas, mientras que únicamente, las más básicas se limitan a ciertas zonas cerebrales.

Además, en este siglo, von Hemholtz consigue medir la velocidad a las que las células nerviosas transmiten impulsos eléctricos [2].

Cabe destacar la importancia del español Santiago Ramón y Cajal, que estableció la base de la neurociencia y por la que recibió junto con el italiano Camillo Golgi, el Premio Nobel de Medicina en 1906. Golgi desarrolló técnicas experimentales de tinción, las cuales Cajal perfeccionó para las demostraciones de sus descubrimientos.

Cajal desarrolló una teoría neuronal denominada *Doctrina de la neurona,* según la cual se establece el hecho de que el cerebro está formado por células individuales refiriéndose a ellas como neuronas (aparece este término por primera vez en 1891 por Wilhelm von Waldeyer). Además, expone que las neuronas interaccionan entre sí, mediante conexionen fijas y en lugares concretos de las células (más tardes denominados sinapsis por Sherrington) y describe la polarización dinámica que tiene lugar desde las dendritas hasta el axón (veremos esto con más detalle en la sección 1.3).

Esta teoría sigue vigente hoy en día y supuso un cambio de pensamiento para muchos científicos de la época, ya que hasta entonces, todas las teorías eran reticulares, es decir, concebían la red neuronal como un conjunto y no consideraban la independencia de las neuronas.

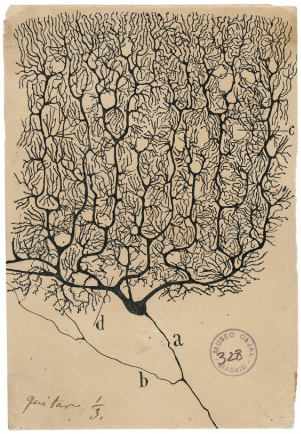

Figura 1.1: Dibujo de la tinción de Santiago Ramón y Cajal a una neurona de tipo *Purkinje* [3].

Debido a esto, muchos investigadores sitúan el inicio de la neurociencia en estas fechas y a Santiago Ramón y Cajal como padre de esta ciencia.

En 1929 Hans Berger inventa el electroencefalógrafo *(EEG)* capaz de medir la actividad eléctrica del cerebro. En 1938, Isidor Rabi descubre la imagen por resonancia magnética *(IRM)*, por la que recibió seis años más tarde el Premio Nobel.

Hodgkin y Huxley describieron un modelo matemático en 1952 relacionado con el potencial de acción y su propagación, con el que también obtuvieron el Premio Nobel junto con Eccles.

En 1962, impulsada por el *Massachusetts Institute of Technology (MIT)*, da comienzo la primera organización de neurociencia, *Neuroscience Research Program*.

En el siglo *XXI* se lanzan varios proyectos, entre ellos los mencionados anteriormente *BRAIN* y *HBP*, y se comienza a dar la importancia que requiere la neurociencia.

1.2. Anatomía básica

A continuación, se presentan conceptos básicos de fisiología necesarios para comprender la importancia del estudio.

La mayoría de animales pluricelulares cuentan con sistema nervioso. Éste está formado por una red compleja de neuronas conectadas y se puede dividir mediante:

- *Sistema Nervioso Central - SNC* (encéfalo y médula espinal)

- *Sistema Nervioso Periférico - SNP* (resto de nervios que no se encuentran en el SNC)

El *encéfalo* es la parte que está contenida en el cráneo y está formado por el *cerebro*, el *cerebelo* y el *tronco encefálico*. El *SNP* conecta el *SNC* con otras partes del cuerpo. El *SNP* envía estímulos a través de neuronas sensoriales al *SNC*, éste las procesa y manda la reacción a las neuronas motoras mediante el *SNP* para que se ejecute la respuesta. A continuación, se ilustra el Sistema Nervioso de un humano en la Figura 1.2.

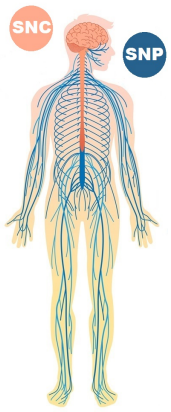

Figura 1.2: Representación del Sistema Nervioso Central - SNC en color rojo y Sistema Nervioso Periférico - SNP en color azul de un humano [4].

El cerebro se divide en distintas áreas que controlan diferentes labores y que están conectadas entre sí. Además, la parte posterior del cerebro tiene *organización contralateral*, es decir, la parte izquierda del cerebro controla el lado derecho del cuerpo y la parte derecha del cerebro controla el lado izquierdo. Esto ocurre en la vista, que principalmente se encuentra en la zona de detrás, en cambio con el olfato esto no ocurre y se denomina *organización ipsilateral*. En seres invertebrados no tiene lugar la organización contralateral.

La parte emocional y cognitiva se encuentra principalmente en el área frontal. Las diversas labores que tienen lugar en los experimentos realizados en estudios neurocientíficos implican numerosas áreas, como veremos en la sección 1.5.

1.2.1. Neuronas

Las neuronas son las unidades básicas del sistema nervioso. Estas células son responsables de procesar y transmitir la información a través de impulsos eléctricos. El cuerpo celular se denomina *soma*, del cual surgen extensiones ramificadas cortas llamadas *dendritas* que son las encargadas de recibir las señales y una extensión que suele ser más larga, denominada *axón*. En la siguiente imagen se presentan las distintas partes de una neurona (Figura 1.3). El tamaño del soma está entre 5 y 135 micrómetros, mientras que el axón puede llegar a medir más de un metro.

La estructura de una neurona, así como la cantidad de dendritas puede variar dependiendo de su especialización. Las neuronas situadas en el cerebro suelen ser de menor tamaño y contener mayor cantidad de dendritas.

Para que nos hagamos una idea de la complejidad de la red neuronal que forma nuestro sistema nervioso, en su extensión es superior a los 150.000 km. Únicamente el cerebro cuenta con más de cien mil millones de neuronas, donde cada una tiene una media de 7.000 conexiones con otras neuronas. Una neurona de la columna vertebral puede conectarse con más de 10.000 neuronas postsinápticas [6].

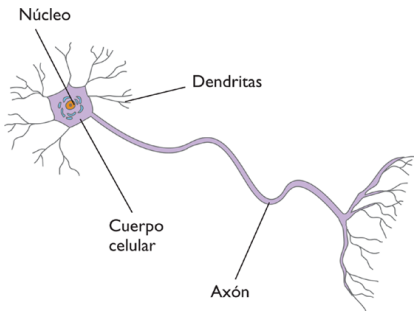

Figura 1.3: Partes de una neurona, entre las que destacan el cuerpo celular o soma que contiene el núcleo, y las ramificaciones que están compuestas por las dendritas y el axón [5].

1.3. Campo receptivo

Según las señales recibidas por las dendritas de una neurona, ésta se inhibe o provoca un impulso eléctrico, también conocido como *potencial de acción* o *spike*. De esta manera, se liberan ciertas moléculas (*neurotransmisores*) desde el extremo del axón de la neurona presináptica hacia el soma o una dendrita de la neurona postsináptica. La región en la que tiene lugar el enlace entre neuronas se denomina *sinapsis*. Normalmente el impulso eléctrico tiene una amplitud de 100 mV y una duración entre 1 y 2 ms.

Se puede ver que la forma del *spike* es siempre similar. Consta de tres fases: *despolarización*, *repolarización* e *hiperpolarización*. En la primera se produce una apertura de canales que dejan entrar iones positivos (por ejemplo Na^+) a la célula o que permite la salida de iones negativos provocando en ambos casos un aumento del potencial. La *repolarización* se corresponde con la vuelta al potencial en reposo. Por último, la *hiperpolarización* tiene lugar por la entrada de cargas negativas o la salida de cargas positivas a la célula [7].

En la Figura 1.4 podemos ver una representación de las distintas fases sombreadas de colores azul, verde y amarillo respectivamente. Cabe mencionar que todas curvas suelen tener la misma forma y altura.

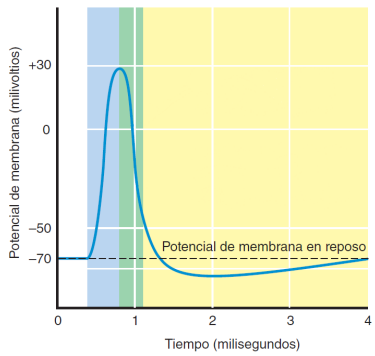

Figura 1.4: Potencial de una neurona durante un *spike*. Las zonas sombreadas se corresponden con las fases. En azul claro tiene lugar la *despolarización*, en color verde la *repolarización* y en color amarillo la *hiperpolarización*. El eje X se corresponde con el tiempo en milisegundos. Imagen modificada de [7].

El potencial de acción tiene un carácter refractario en el cual no se pueden producir dos *spikes* de manera simultánea. Cuando se alcanza el máximo de *spikes* por segundo, es decir, la tasa de disparo (concepto que veremos más tarde) es máxima, se dice que la neurona está *saturada*. El periodo refractario absoluto del potencial de acción tiene lugar durante las fases de *despolarización* y *repolarización*. En caso de que hubiera un estímulo muy intenso en la fase de *hiperpolarización* sería posible producir un segundo potencial de acción y se dice que tiene periodo refractario relativo [7].

Dado que esto ocurre en un periodo de tiempo muy pequeño, se suele representar los potenciales de acción discretamente, mediante *spike trains* ya que los *spikes* están separados de manera que no es posible producir uno cuando todavía no ha terminado el anterior.

Se suele dibujar el eje horizontal que representa el tiempo y distintas barras verticales que indican cuándo se ha producido cada *spike*. Esto se puede ver en la Figura 1.3, en ella se muestran los distintos comportamientos de dos neuronas frente a diversos estímulos. En la imagen superior la inclinación de la barra situada a la izquierda es lo que se modifica, mientras que en la inferior se muestran cuatro barras en movimiento según la dirección que indican las flechas y orientadas de manera diferente (vertical/horizontal). La zona sombreada de gris oscuro representa el espacio de tiempo en el que se muestran los estímulos. Se ve cómo la neurona representada en la imagen superior aumenta la cantidad de *spikes* conforme la barra de la izquierda adopta la posición vertical. En la imagen inferior la neurona presenta más *spikes* cuando las barras están situadas horizontalmente y se mueven hacia abajo.

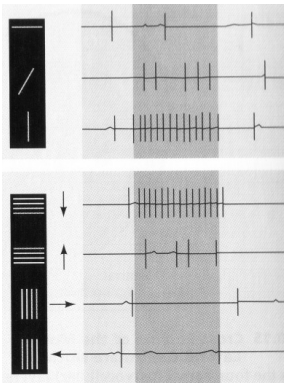

Figura 1.5: *Spike trains* de dos neuronas cuando un sujeto observa diversos estímulos. La zona sombreada de gris oscuro representa el tiempo en el que se muestra el estímulo. En la parte superior la diferencia de los estímulos es la inclinación de la barra. En la parte inferior, la variación se encuentra en la posición de cuatro barras (vertical/horizontal) y el desplazamiento indicado mediante las flechas.

Como se indica anteriormente, miles de *spikes* son producidos cada milisegundo en poca cantidad de espacio. Se considera como *tasas de disparo* o *firing rates* de una neurona a la cantidad de *spikes* por segundo. Y van a ser estos los datos con los que vamos a desarrollar este trabajo.

Las neuronas son células muy especializadas que responden a estímulos muy concretos. El tipo o la zona de estímulo que modifica la tasa de disparo de una neurona se denomina *campo receptivo.* Por ejemplo, el campo receptivo de una neurona del sistema visual es el área de la retina que produce modificaciones en su actividad.

En resultados experimentales [8] es común observar como para cada comportamiento de una neurona, encontramos otra que tiene respuesta opuesta. Cuando dicha neurona se estimula, tiene lugar a la vez un proceso de inhibición para otra.

Una alta actividad neuronal no implica mejor funcionamiento del cerebro como se ha pensado durante mucho tiempo. De hecho, un exceso de actividad puede tener terribles consecuencias, como ocurre en ataques epilépticos.

La ciencia y la tecnología han evolucionado de manera que es posible cuantificar los *spikes* de neuronas aisladas a través de electrodos intracelulares que miden la diferencia de potencial entre el interior de la neurona y su alrededor.

La proximidad física de las neuronas no está relacionada directamente con el mismo campo receptivo, a pesar de que hay zonas donde sí lo están.

En la parte superior de la Figura 1.6, se puede ver la diferencia del comportamiento de una neurona según el contraste de una imagen. En la parte superior aparecen dos *spike trains* en los cuales el eje X se corresponde con el tiempo y las barras verticales indican el tiempo en el que se producen los *spikes* cuando el sujeto mira la imagen situada a la izquierda. En el gráfico de la parte inferior de la imagen,

se puede ver cómo varía la tasa media de disparo según el porcentaje del contraste. Además, se ha producido un ajuste sigmoidal, muy frecuente en estudios de tasas de disparo. En el análisis, apenas se ve actividad cuando el contraste es bajo, en cambio, aumenta cuando el contraste se incrementa. ¿Qué deducciones podemos hacer del gráfico resultante?

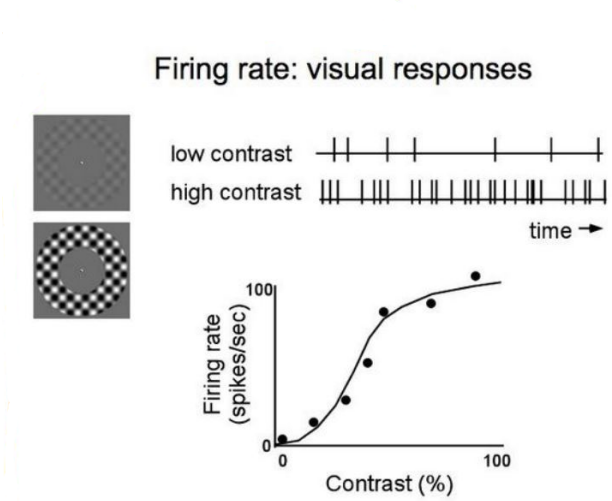

Figura 1.6: *Spikes train* y tasa de disparo de una neurona en respuesta a una imagen cuyo contraste se modifica. En la parte superior, el eje X corresponde al tiempo y las barras verticales indican los tiempos en los que se producen los *spikes* cuando el sujeto observa la imagen situada a la izquierda. En la gráfica inferior, el eje X se corresponde con el contraste de la imagen que observa el sujeto y el eje Y es la tasa de disparo de la neurona. Los puntos son medidas y se ha añadido un ajuste sigmoidal.

Por lo que surgen diversas preguntas, ¿qué información contienen las neuronas y cómo está relacionada con las tasas de disparo? ¿cómo codifican esa información? ¿y cómo la decodifican otras neuronas? ¿se pueden resolver estas preguntas con las tasas de disparo?

Para muchas de estas cuestiones todavía no hay respuestas. En este trabajo vamos a analizar y reproducir un algoritmo que ayude a saber si, dada una muestra de neuronas de la cuales conocemos su tasa de disparo frente a distintos estímulos, codifican un conjunto discreto de variables (*categorical encoding*) o por el contrario, un conjunto arbitrario de variables (*categorical free*). Además, en el caso de obtener un conjunto discreto, si se proponen varias variables como posibles, nos va a permitir decir cuáles de ellas codifica.

1.4. Tareas de decisión

Para estudiar las respuestas de las neuronas frente a diversos estímulos, podemos restringir el entorno de modo que se pueda controlar y cuantificar el medio. Se va a proceder a través de tareas en las que se especifican de manera muy concreta el exterior y ver así cómo se modifica el comportamiento de las neuronas. Reduciendo las máximas distracciones posibles.

A continuación, nos vamos a centrar en tareas de decisión. En las cuales, se presentan al sujeto varios estímulos y éste debe elegir entre uno de ellos. El objetivo es estudiar el efecto que los estímulos tienen en la actividad neuronal.

El cerebro tiene que realizar una serie de procesos entre los que, dependiendo de la tarea, se encuentran: recibir y procesar la información, guardarla en memoria, compararla y realizar la elección. Un buen diseño experimental permitirá aislar estos procesos en el tiempo y analizarlos separadamente.

Este tipo de tarea se denomina de selección forzosa de dos alternativas (*2AFC - Two-alternative forced choice*) cuando son dos los estímulos que se exponen. Veremos un ejemplo concreto y detallado en el Capítulo 2.

Además, según la estructura de la tarea y el objetivo, se suele dividir el experimento en diferentes ventanas de tiempo, también denominadas *time windows*. Estas ventanas pueden solaparse, se les puede aplicar medias móviles o incluso ambas opciones si se considera oportuno. Esto permite realizar el estudio en distintas secciones del experimento. En tareas de toma de decisión es frecuente estudiar la ventana de tiempo tras la muestra de los estímulos y tras la respuesta del sujeto. Pero, en algunas tareas puede interesar el comienzo del ensayo o el momento previo a la elección. Cabe mencionar, que también hay que considerar el tiempo de reacción, así como el tiempo de respuesta, que suele ser de 10 ms. Este tiempo sería el correspondiente a la duración desde que se muestra el estímulo hasta que llega al cerebro o el tiempo que tarda el impulso de ir desde el cerebro a la zona motora, que dependerá del tipo de tarea.

1.5. Áreas cerebrales relacionadas con tareas de toma de decisión

En la parte externa del cerebro se encuentra la corteza cerebral que está formada principalmente por masa gris, que a su vez está compuesta mayormente por cuerpos celulares y dendritas. El cerebro consta de cinco lóbulos (frontal, parietal, temporal, occipital, ínsula) situados en dos hemisferios. Cada área del cerebro tiene unas funciones principales y se asocian con distintas partes del cuerpo, con emociones, memoria o características funcionales del cuerpo.

En las tareas de toma de decisión las neuronas juegan diferentes roles. En el proceso completo de la tarea, el cerebro debe realizar distintas labores en las que intervienen diversas áreas del cerebro. Una tarea de decisión genérica se puede dividir en las siguientes subtareas (aunque puede que en un diseño experimental concreto no tenga lugar alguna de ellas).

- Recibir la información
- Guardar en memoria
- Comparar
- Razonar una elección
- Realizarla

Se expone a continuación un ejemplo concreto, en el que se puede observar la variabilidad de las áreas implicadas en tareas de toma de decisión.

1.5.1. Tarea de discriminación vibrotáctil

Este experimento fue diseñado por el Dr. Ranulfo Romo. Se realizan dos vibraciones en el dedo del sujeto y después tiene que elegir la que tiene mayor frecuencia. Para ello, primero se realiza el primer estímulo, que vamos a denotar como f_1. Al cabo de 3 segundos se produce la segunda vibración, f_2, forzando de esta manera a guardar en memoria la primera frecuencia. Después se comparan ambas vibraciones y se indica la elección mediante uno de los dos botones, tomando la decisión de cuál de las vibraciones tiene la frecuencia más alta. Uno de ellos indica $f_1 < f_2$ y el otro $f_2 < f_1$. Además, también se añadió un retraso entre la aparición del segundo estímulo y la ejecución de la respuesta, para forzar a que mantengan la información de la respuesta en memoria.

Los estímulos son las vibraciones con las respectivas frecuencias que se realizan en el dedo. Este tipo de tarea se denomina de discriminación vibrotáctil y forma parte del campo de la percepción somatosensorial.

LaMotte y Mountcastle comienzan a estudiar estímulos vibrotáctiles [9]. Durante bastante tiempo, se analizan tareas similares a la descrita, con mínimas variaciones en las que se estudia las diferentes

áreas implicadas por separado. No es hasta 2010, cuando se realiza el experimento a macacos obteniendo los datos de las diferentes áreas ilustradas en la Figura 1.7 de manera simultánea.

Las áreas implicadas son *Primary somatosensory cortex (S1), secondary somatosensory cortex (S2), medial premotor cortex (MPC), prefrontal cortex (PFC), ventral prefrontal cortex (VPC), dorsal premotor cortex (DPC),* y *primary motor cortex (M1)*. En la elección de la posición de los electrodos para la recogida de datos, se tiene en cuenta el carácter contralateral o ipsilateral de las zonas implicadas.

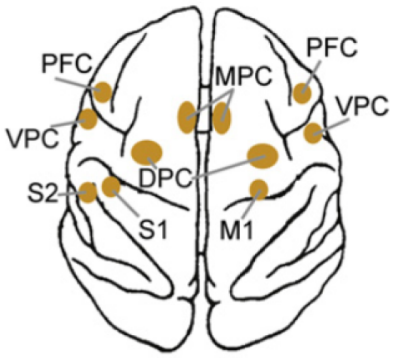

Figura 1.7: Áreas estudiadas de un cerebro de un mono durante el experimento mientras realiza tareas de toma de decisión de discriminación vibrotáctil en [10]. Las áreas implicadas son *Primary somatosensory cortex (S1), secondary somatosensory cortex (S2), medial premotor cortex (MPC), prefrontal cortex (PFC), ventral prefrontal cortex (VPC), dorsal premotor cortex (DPC)*, y *primary motor cortex (M1)*.

Mediante un análisis de regresión lineal multivariante para cada área según el cual se tiene

$$r(t) = a_0(t) + a_1(t) f_1 + a_2(t) f_2,$$

donde $r(t)$ es la tasa de disparo de una neurona, se concluye que S1 es sensible a los estímulos, ya que presenta actividad relacionada directamente cuando estos se presentan y son independientes en los periodos libres de vibraciones.

Para el resto de áreas, a pesar de que tienen diferente comportamiento, deducen que hay una transmisión de la información hacia el lóbulo frontal encargado de ejecutar la respuesta.

Capítulo 2

Estudio de tasas de disparo

En el capítulo anterior hemos visto el efecto que ciertos estímulos pueden provocar en el comportamiento de las neuronas. No está claro que el estudio únicamente de las tasas de disparo nos vaya a resolver todas las dudas, pero sí se puede observar que tienen un elevado grado de implicación y por ello, su estudio es la finalidad de este trabajo.

Una parte de la neurociencia importante a investigar es el hecho de cómo codifican la información las neuronas. A través de las tareas podemos alterar y controlar el entorno, y por tanto las variables que pueden llegar a codificar. De manera que si variamos únicamente una variable y observamos que la tasa de disparo se modifica notablemente, es esperable considerar que esa neurona codifique dicha variable.

Esto se complica cuando en el entorno hay muchas variables y además hay correlación entre ellas. Por eso, es importante considerar una tarea sencilla, en la que interfieran el menor número posible de variables.

En este capítulo vamos a estudiar, a través de las tasas de disparo, la propiedad de que una neurona sea categórica o no a través de procedimientos estadísticos.

2.1. Categórica vs no categórica

Una neurona se considera *categórica* si codifica un conjunto discreto de variables. En caso contrario, se dice que es *no categórica* o que es *category free*.

Esta cuestión es complicada puesto que neuronas que se encuentran cerca físicamente, pueden codificar la información de manera diferente y las variables a considerar pueden estar correladas. Además, una neurona puede codificar variables distintas en periodos distintos. Todo esto implica mayor dificultad a la hora de determinar si una neurona es categórica o no.

Como vimos en el Capítulo 1, el cerebro está dividido en diferentes áreas que no son independientes entre sí, es decir, que están conectadas. A pesar de eso, distintos tipos de acciones producen una actividad mayor en ciertas regiones del cerebro.

Las tareas de decisión, que son en las que nos vamos a centrar, tienen su actividad principalmente en el área denominada *orbitofrontal cortex - OFC*, situada en la parte frontal en primates y humanos. El área *OFC* se relaciona con toma de decisiones, emociones y personalidad entre otras características [11].

Podemos ver una representación del *OFC* en la Figura 2.1 correspondientes al cerebro humano, al de un macaco y una rata respectivamente.

A pesar de la variabilidad de las áreas del cerebro implicadas en tareas de decisión comentadas en 1.5, las posibles variables codificadas en la toma de decisiones estarían en *OFC*, ya que se ha observado que lesiones en esta zona del cerebro en primates alteran los resultados [12] y también en imágenes cerebrales humanos [13].

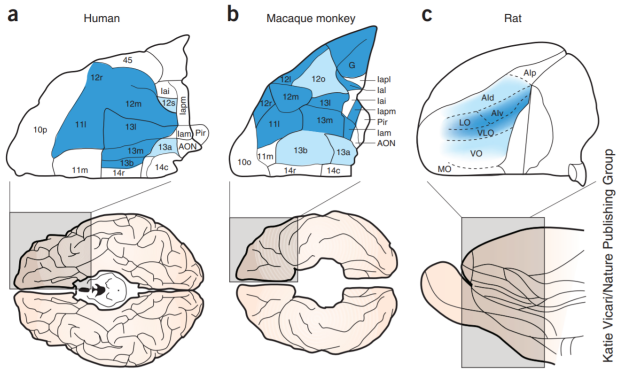

Figura 2.1: Las zonas sombreadas en azul representa el *OFC* en cerebro humano, en el de un macaco y una rata respectivamente. Imagen obtenida en [11].

Alternativamente, en la literatura podemos encontrar diversos estudios que muestran un comportamiento no categórico de neuronas en áreas prefontales de macacos [14]. Como podemos notar, no hay un criterio claro sobre el tema, numerosos estudios se contradicen en la cuestión de la clasificación de las neuronas.

Esta cuestión no es únicamente relevante en las áreas de toma de decisión, por ejemplo lo autores de [15] determinan que las neuronas del *parietal córtex* relacionadas con estímulos visuales son no categóricas.

Frente a la diversidad de resultados en la literatura, se expone en este capítulo un método que analiza el comportamiento de las neuronas. Cabe mencionar que este procedimiento es independiente de la región de la que se obtengan los datos.

Para ello, primero vamos a describir una tarea concreta y dar conceptos propios de este área. De esta manera, contextualizaremos y entenderemos mejor la aplicación y el desarrollo del algoritmo.

2.2. Descripción de la tarea, datos y resultados

Los experimentos mencionados en [16], consisten en la realización de toma de decisiones en dos monos rhesus mediante una tarea mostrada en la Figura 2.2.

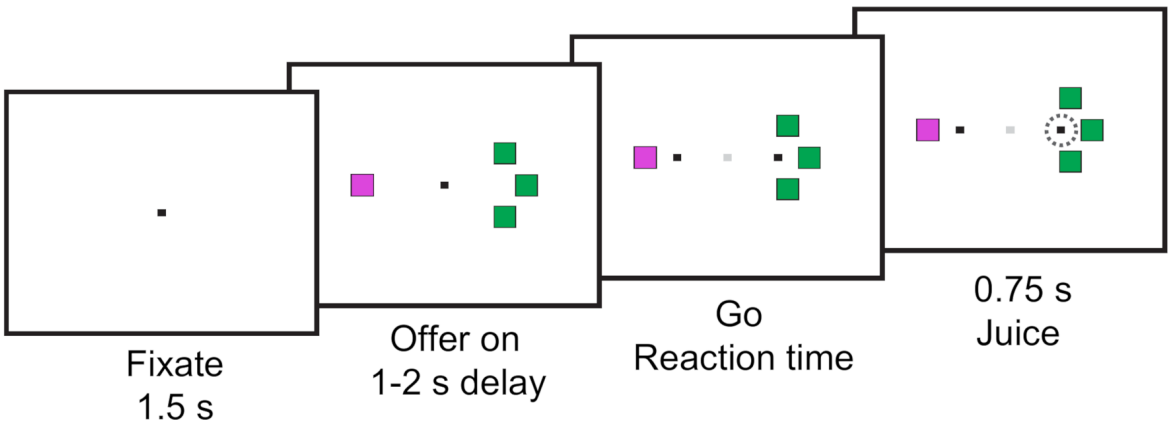

Figura 2.2: Descripción de la tarea [16]. Mantener mirada fija durante 1500ms. Se muestran dos conjuntos de cuadrados a cada extremo, representando el sabor del zumo según el color y la cantidad según el número de cuadrados. Tiempo de reacción y a través de movimiento ocular la posterior elección.

La tarea a desarrollar es una tarea que se denomina de elección económica. Consiste en la presenta-

ción de dos zumos *A* y *B* en diferentes cantidades y una posterior elección de uno de ellos. En [17] se realiza un estudio mediante un modelo de inferencia logística en el que se concluye que hay preferencia de *A* sobre *B*.

Un tipo de prueba o *trial type* que se represente mediante [1A:3B,B] significa que se ha ofrecido 1 cantidad de zumo A, 3 de zumo B y la elección ha sido el zumo B. Se realizan varias sesiones en las cuales cada oferta se realiza al menos 20 veces.

Cada ensayo procede de la siguiente manera. Primero, se observa fijamente una pantalla. Después de 500 ms se presentan dos conjuntos de cuadrados de dos colores diferentes, uno en cada extremo. Estos conjuntos representan los zumos con las cantidades correspondientes al número de cuadrados. Al cabo de 1-2 segundos, indican, a través de movimiento ocular, la elección deseada, manteniendo durante 750 ms.

Hay una gran variabilidad en los tipos de pruebas para evitar tendencias en las elecciones. Debido a los múltiples cambios en las ofertas se pueden considerar los ensayos independientes. Un análisis más detallado de este tema denominado *choice hysteresis* lo podéis encontrar en [17].

Las variables consideradas a estudiar fueron las 10 siguientes

- Cantidad de zumo A ofrecido
- Cantidad de zumo B ofrecido
- Cantidad de zumo elegido
- Cantidad de zumo no elegido
- Cantidad de zumo A elegido
- Cantidad de zumo B elegido
- Cantidad del elegido - cantidad no elegido
- Cantidad no elegido / cantidad del elegido
- Número elegido
- Zumo elegido (1 si elige A, 0 si elige B)

Analizamos la actividad neuronal según 4 ventanas de tiempo diferentes, mostradas en la Figura 2.3. Las dos primeras se corresponde a los momentos previos y posteriores a la oferta, mientras que las dos últimas pertenece a la entrega del zumo. Además, entre estos dos grupos, ocurre un tiempo aleatorio, que depende del sujeto.

La cantidad de neuronas que se comportan de manera diferente o incluso opuesta en distintas *time windows* es insignificante [17].

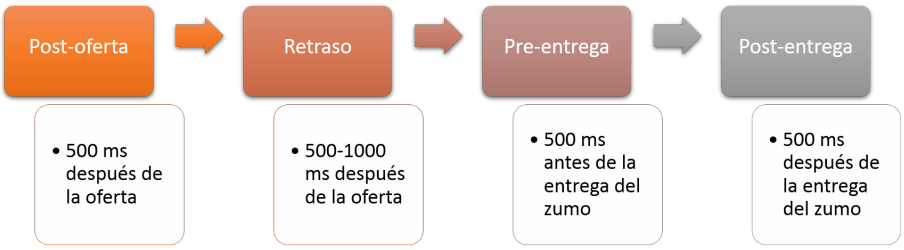

Figura 2.3: Ventanas de tiempo de la tarea. La primera se considera 500 ms después de mostrar la oferta. La segunda entre 500 y 1000 ms tras la oferta. La siguiente es 500 ms antes de entregar el zumo elegido y la post-entrega se considera 500 ms después.

Previo a la aplicación del algoritmo se realiza un estudio en el que se analizan las neuronas relacionadas con la tarea y se agrupan por comportamiento.

2.3. Preparación de los datos

Es importante considerar el hecho de que puede haber neuronas que no codifican ninguna variable relacionada con la tarea. Por ello, por cada ventana de tiempo y cada neurona se realiza un Test Anova

(ver 3.1) con umbral $p < 0{,}001$ con respecto a los diferentes *trial types*. De esta manera, se comparan las medias para cada *trial type*. La hipótesis nula H_0 es que todas las pruebas tienen la misma media y por tanto la tasa de disparo es independiente de la tarea. Si se obtiene un p-valor por debajo del umbral, entonces se tiene que hay evidencia para rechazar la hipótesis nula H_0 de que tienen la misma media. Luego, hay al menos un *trial type* para el que la neurona se comporta de manera diferente y por lo tanto se considera que es una candidata a estar relacionada con la tarea.A este grupo se denomina *task related neurons*.

Este no es el único procedimiento para obtener dichas neuronas. En la literatura, también es frecuente realizar un estudio en el que se analiza para cada neurona la media de los *trial types* en dos ventanas de tiempo diferentes, usualmente antes de empezar la tarea y durante el estímulo. De manera que si obtenemos que tiene media diferente, la neurona se considera relacionada con la tarea puesto que ha modificado su comportamiento.

Por otro lado, en cada sesión, no se realizan los mismos *trial types.* Los grupos en los que se puede dividir los datos, de manera que cada grupo tenga los mismos *trial types* y se comporten de manera similar se denominan *pools*.

Para poder aplicar el algoritmo es necesario que sean los mismos *trial types*. Por ello, el procedimiento se realiza no sólo para cada ventana de tiempo, sino también para cada *pool.*

Las pruebas realizadas en cada sesión son diferentes entre sí. Al analizar las sesiones que tienen los mismos *trial types* se obtiene que 5 de los 6 grupos son bimodales. Las neuronas de un mismo *pools* tienen que tener el mismo comportamiento, luego se decide dividir los grupos en los que hay bimodalidad en dos. Además también se eliminan datos atípicos detectados mediante el rango intercuartílico. Finalmente, se establecen 9 *pools* con los mismos *trial types* y valores relativos similares.

En un mundo idealizado, lo mejor sería que todos los datos vinieran de una misma sesión para así evitar las variaciones entre sesiones y minimizar errores. Pero no es física ni éticamente posible, debido a la dificultad en la obtención de los datos.

El objetivo es relacionar un grupo de neuronas con similar comportamiento con variables relacionadas con la tarea. Hay neuronas cuyo comportamiento es opuesto a una variable. En ese caso, codifica dicha variable pero de manera contraria. Es por esto que se añaden los puntos opuestos o *mirror points*, para así agrupar las neuronas que representan la misma variable independientemente del signo. Aunque esto no implica directamente que haya un número par de *clusters.*

2.4. Algoritmo

Presentamos aquí el procedimiento elaborado en [16]. Sus objetivos son, dada la importancia y la falta de procesos efectivos en la materia, desarrollar una serie de procedimientos que permitan esclarecer si una neurona es categórica o no. Este proceso no tiene en cuenta las variables y finalmente, si resulta categórica, compara cuantitativamente una serie de variables propuestas, independientemente de si dichas variables están correladas.

Para ello, se considera el espacio hiper-dimensional definido por los tipos de prueba de la tarea (*trial types*), de manera que el valor en cada dimensión es la media de la tasa de disparo de una neurona para cada *trial type*. Mediante la tarea modificamos el exterior, de esta manera controlamos la información que le puede llegar a la neurona. Se considera la media de la tasa de disparo de cada *trial type* para cada neurona en cada ventana de tiempo. Así, cada célula se corresponde a un punto de ese hiper-espacio. Además, las posibles variables codificadas también se encuentran en un punto del hiper-espacio.

Con el objetivo de facilitar el estudio de este algoritmo, consideramos un caso simplificado en dimensión 2, es decir, vamos a tener en cuenta únicamente dos *trial types*, para así poder hacer una representación gráfica de los pasos realizados. Además, se ha resaltado una neurona en concreto para exponer el procedimiento. En la Figura 2.4 podemos ver los datos correspondientes a 2 *trial types* del *pool* 1. El concepto de *pool* se ve más adelante en la subsección 2.3. De manera más precisa, las ofertas que hemos elegido son [1A:0B] y [3A:1B] (ver sección 2.2 para la descripción de la tarea).

Una vez disponemos de las tasas de disparo en el formato deseado, los centramos y normalizamos

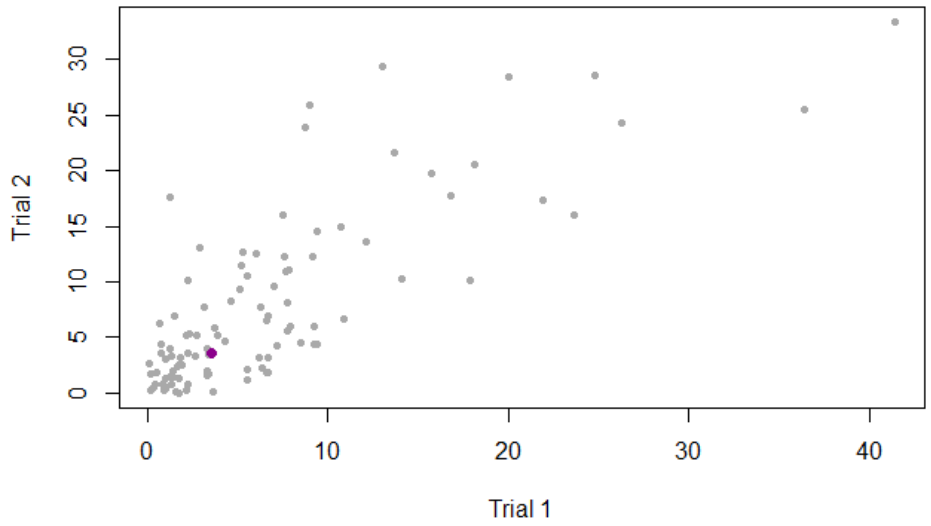

Figura 2.4: Datos correspondientes a los *trial types* 1 y 2 del *pool* 1. Cada punto representa la tasa de disparo media de una neurona según el *trial type*. Se ha destacado la primera neurona del conjunto de datos.

para obtener los datos situados en la hiper-esfera de radio 1. En el ejemplo quedarían los puntos en una circunferencia puesto que estamos considerando el caso de dimensión 2, tal y como se muestra en la Figura 2.5

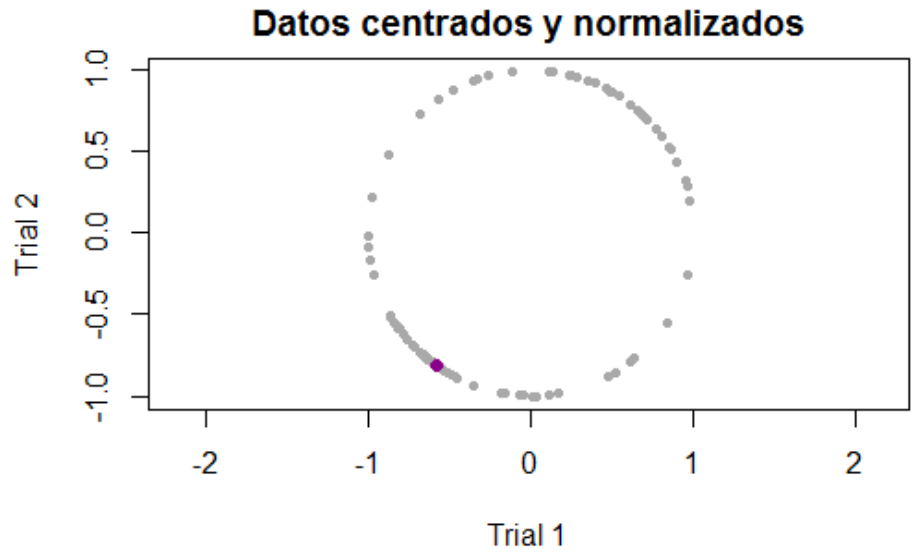

Figura 2.5: Datos centrados y normalizados correspondientes a los *trial types* 1 y 2 del *pool* 1. Se ha destacado la primera neurona del conjunto de datos.

Así, la cuestión de saber si una representación neuronal es categórica se traduce a un problema de clasificación (*clustering*), que se resuelve mediante el método *spherical k-means* que desarrollaremos en profundidad en el Capítulo 3.

Un problema con el que nos encontramos es el que el número de particiones no es conocido, por lo que se realiza *spherical k-means* para varios valores. Este método da señales acerca de la propiedad

de que sea categórica. Una forma de validar un método de clasificación es mediante *silhouette plot*. Muestra el ajuste por cada grupo de la clasificación, de manera que cuanto mayores son los valores, mejor es el ajuste de ese *cluster*. Valores negativos indican que hay elementos de ese grupo que están más cerca de otro *cluster* que del que han sido asignados. Valores próximos a 1 indican que hay un buen ajuste de los datos a los clusters y valores próximos a 0 muestran que hay datos que están a la misma distancia de dos clusters. Véase 3.3 para más detalle.

Además, a través de un análisis con datos sintéticos categóricos, determinan que los gráficos resultantes son convexos. Mientras que para datos uniformes son valores positivos cercanos a 0 y los gráficos cóncavos. Con este criterio, se diferencia entre categórico o no categórico.

En la Figura 2.6 mostramos los *silhouette plots* respectivos al *pool 1* teniendo en cuenta nueve *trial types* para los casos de 6, 7, 8 y 9 *clusters*. Para cada caso se indica la media de los valores $s(i)$. Se observa convexidad en los datos y pocos valores negativos. Estos resultados son consistentes con los obtenidos en [16].

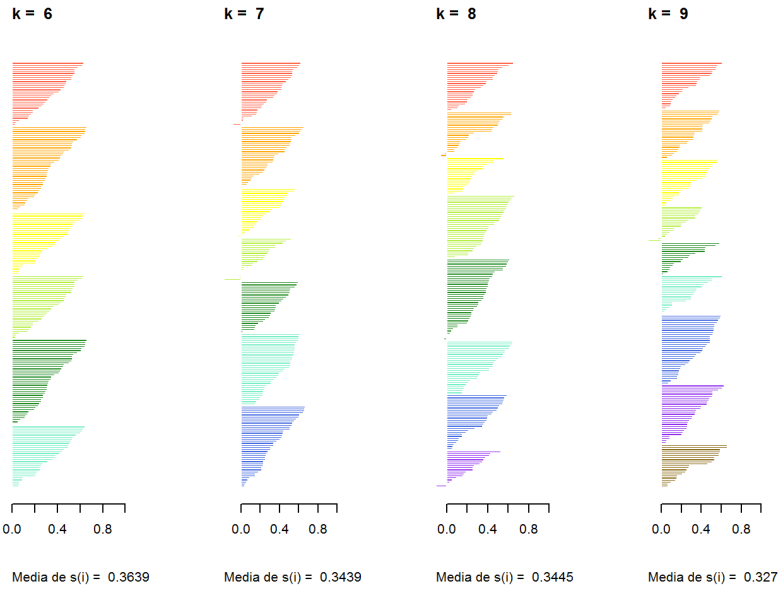

Figura 2.6: *Silhouette plots* para neuronas del *pool* 1 considerando todos los *trial types*. Se observa convexidad y pocos valores negativos. Se indica el número de *clusters* considerados así como la media de los valores $s(i)$.

Pero *spherical k-means* no nos indica cuáles son las variables codificadas por cada *cluster*. Para ello, aplicamos simultáneamente otro procedimiento de clasificación denominado *variable-centroid*. Este método sí utiliza las variables propuestas. Dado un número de *clusters* se eligen aleatoria e iterativamente dicha cantidad de variables que constituyen los centros. De esta manera, cada neurona corresponderá al grupo del centro que más cerca se encuentre. Recordemos que nos hemos situado en el hiper-espacio en el que cada eje se corresponde con un tipo de prueba y por tanto cada posible variable está en ese hiper-espacio.

Una de las principales ventajas de normalizar las tasas de disparo es que las variables pertenecen a la misma escala y por tanto, tiene sentido utilizarlas y calcular la partición mediante *variable-centroid*. En el ejemplo que hemos visto anteriormente, si tomamos las 10 variables descritas en 2.2 podemos ver como efectivamente forman parte de la circunferencia en color amarillo en la Figura 2.7. Debido a que hemos considerado únicamente dos *trial types*, observamos como algunas variables se solapan, puesto que toman el mismo valor para ambas ofertas. Aquí podemos ver la importancia de una correcta planificación de la tarea.

Por lo que obtenemos dos particiones de las respuestas de las neuronas: según *spherical k-means* y según *variable-centroid*. *Spherical k-means* no utiliza las variables propuestas, por lo que el estudio

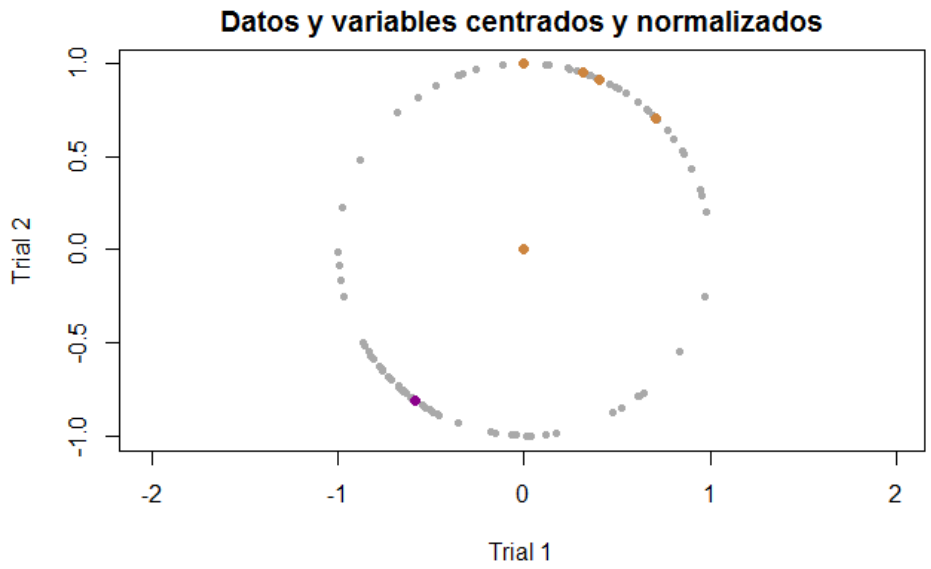

Figura 2.7: Datos centrados y normalizados correspondientes a los *trial types* 1 y 2 del *pool* 1. Se ha destacado la primera neurona del conjunto de datos en color morado y variables propuestas que se han superpuesto en color amarillo.

de la propiedad categorica no depende de estas. Notar además, que el objetivo, en el caso de que la población sea categórica, es determinar las variables que mejor describen el comportamiento de las neuronas. Además, el número de clusters no es conocido, y esto deriva en que tengamos que estudiar el ajuste para varios valores y ver cuál es el mejor.

Por ello, utilizamos como medida de similitud *Adjusted Mutual Information - AMI*, descrito en 3.5.2, y aplicamos repetidamente los métodos mencionados hasta ahora variando la cantidad de *clusters* así como el subconjunto de variables. Además se puede ver que una población sin propiedad categórica muestra valores de *AMI* próximos a 0 [16].

A continuación, mostramos los valores *AMI* obtenidos para el *pool* 1 considerando nueve *trial types*. Para calcular los distintos valores, hemos ido variando la cantidad de *clusters* para el *spherical k-means* y el número de variables a considerar para el método *variable centroid clustering*.

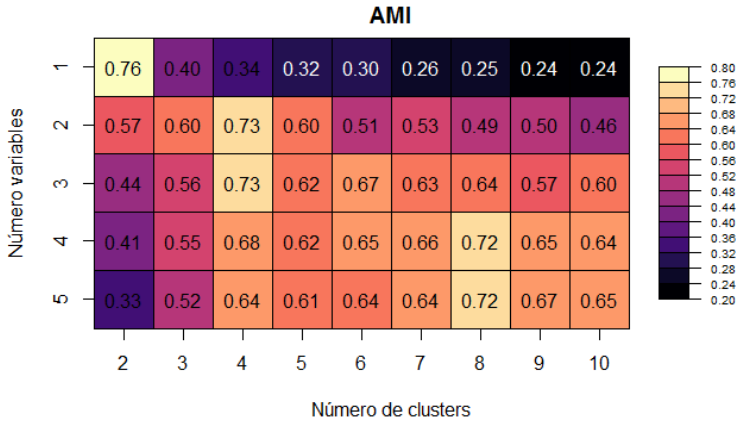

Figura 2.8: Valores AMI obtenidos para *pool* 1 considerando todos los *trial types* variando el número de *clusters* así como el de variables para los procedimientos de clasificación *spherical k-means* y *variable centroid*.

Asimismo, para realizar un mayor estudio de *AMI*, se utiliza el *análisis de Jackknife*, que muestra la variación de los distintos valores *AMI* y mide el error estándar.

2.5. Resultados

Al realizar los *silhouette plots* se observa convexidad, lo que implica un comportamiento categórico de los datos. Los resultados para la ventana *post-entrega* y el *pool* 1, se muestran en la Figura 2.9 (A-G).

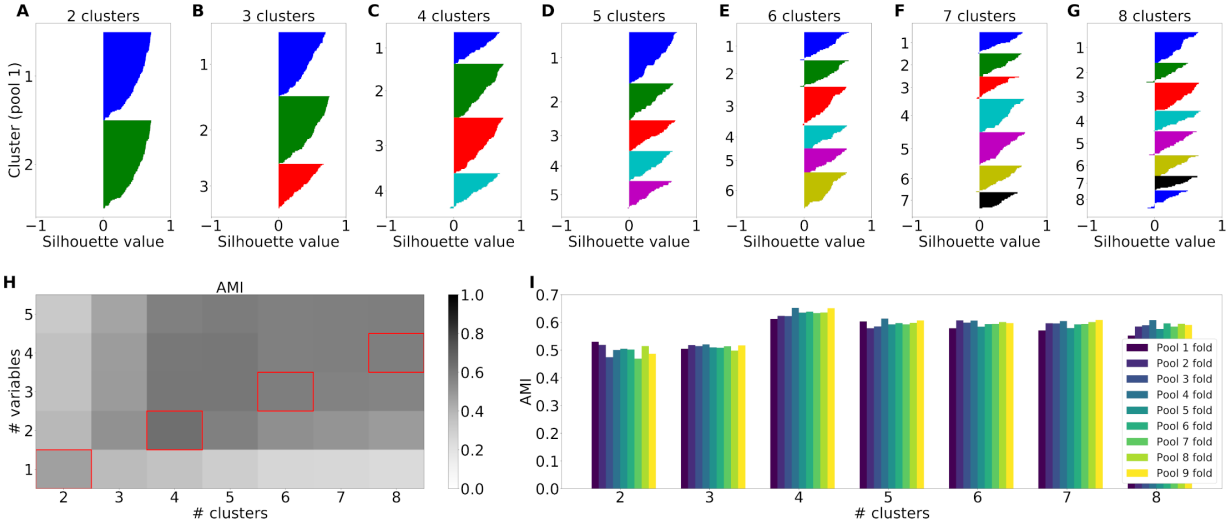

Figura 2.9: Resultados de aplicar el algoritmo a los datos correspondientes a la ventana *post-entrega*.(A-G) *Silhouette plots* del *pool* 1. (H) Valores *AMI* en función del número de *clusters* para la partición obtenida mediante *spherical k-means* y el número de variables que forman la otra partición sobre todos los *pools*. (I) Valores de *Jackknife* sobre cada *pool* para cada número de *clusters*. Se observa convexidad y altos valores de *AMI* para 4 *clusters* y 2 variables para todos los *pools*. Imagen obtenida en [16].

Se puede observar convexidad en los *silhouette* plots, que junto con valores altos de *AMI* indican estructura categórica. Además, para todos los *pools* se obtiene mejores valores para el caso de cuatro *clusters* y dos variables. Se obtienen resultados similares para el resto de ventanas de tiempo.

Aplicando iterativamente *AMI* para comparar las posibles variables y eligiendo la combinación que maximiza ese valor, se observa que,

- Para 2 variables y 4 *clusters* las variables elegidas son “Cantidad de zumo *A* elegido" y “Cantidad de zumo *B* elegido".
- Para 3 variables y 6 *clusters* se obtienen “Cantidad de zumo *A* ofrecido", “Cantidad de zumo *B* ofrecido" y “Zumo elegido $(1A/0B)$".
- Para 4 variables y 8 *clusters* las variables son “Cantidad de zumo *A* ofrecido", “Cantidad de zumo *B* ofrecido", “Zumo elegido $(1A/0B)$" y “Cantidad de zumo".
- Para 5 variables y 10 *clusters* son “Cantidad de zumo *A* ofrecido", “Cantidad de zumo *B* ofrecido", “Zumo elegido $(1A/0B)$", “Cantidad de zumo" y “Cantidad no elegido / cantidad del elegido".

Estas deducciones se producen para cada uno de los *pools*.

Se encuentra consistencia en los resultados, puesto que añadir un *cluster* conlleva a agregar una variable y no modifica las propuestas.

Las conclusiones establecidas en [16] así como las variables escogidas concuerdan con los resultados obtenidos en la literatura [18] [19].

Por otro lado, también se observa que si los *trial types* no están ordenados, no es posible realizar correctamente este estudio. Para ello, se crean unos datos mezclados a partir de los dados y así, de esta manera, se pierde la estructura categórica. Como resultado se obtiene los gráficos de la Figura 2.10 donde se puede observar los *silhouettes* cóncavos, los valores *AMI* muy pequeños y homogéneos y las cantidades de *Jacknife* también son reducidas.

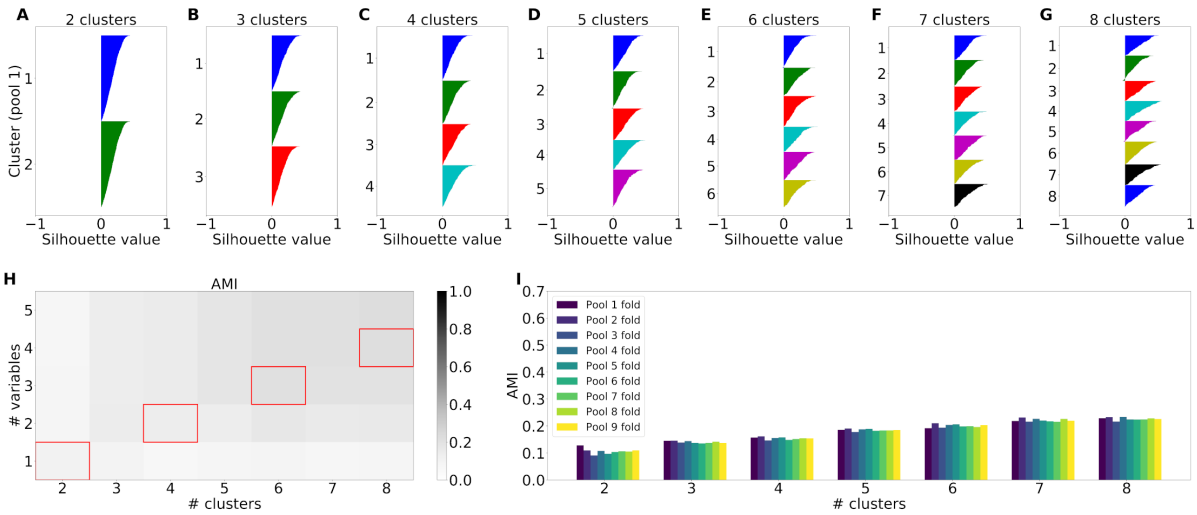

Figura 2.10: Resultados de aplicar el algoritmo a datos mezclados con falta de estructura categórica. .(A-G) *Silhouette plots* del *pool* 1. (H) Valores *AMI* en función del número de *clusters* para la partición obtenida mediante *spherical k-means* y el número de variables que forman la otra partición sobre todos los *pools*. (I) Valores de *Jackknife* sobre cada *pool* para cada número de *clusters*. Imagen obtenida en [16].

Capítulo 3

Métodos

Vamos a estudiar y describir los métodos utilizados en el procedimiento del Capítulo 2. Veremos sus aplicaciones así como daremos alternativas que podemos encontrar en la literatura.

3.1. Test Anova

Sean n poblaciones independientes con distribución normal y misma varianza, este método permite determinar si todas tienen la misma media o, en caso contrario, al menos una de ellas tiene distinta media.

Se realiza un contraste de hipótesis, donde la hipótesis nula, H_0 es la igualdad de medias y en la hipótesis alternativa hay al menos un par de poblaciones que tienen distinta media.

$$\begin{cases} H_0: \ \mu_{trial_1} = \mu_{trial_2} = \cdots = \mu_{trial_n} \\ H_1: \ \text{al menos un par es distinto} \end{cases}$$

En el caso que contemos con únicamente dos grupos, es decir, tengamos $n = 2$ se suele utilizar **t-test** que trabaja con la distribución *t-Student*.

El inicio de los estudios de comportamiento de neuronas siempre comienzan analizando qué neuronas están relacionadas con la tarea. En ocasiones se usa este método comparando las medias de las tasas de disparo. En [16] se ha aplicado este test, con un umbral $p = 0{,}001$, para cada ventana y neurona modificando el *trial type*, es decir, n es la cantidad *trial types* diferentes que se realizan.

Debido a que para aplicar este método es necesario asumir normalidad, es frecuente en la literatura comparar la media para cada neurona y *trial type* instantes antes y después del estímulo. De esta manera, si se obtiene que tienen distinta media la neurona ha cambiado su comportamiento y por lo tanto está relacionada con la tarea.

En la práctica, el inconveniente de utilizar este método es el hecho que hay que asumir normalidad en los datos. Si se tiene un número pequeño de muestras es difícil realizar dicha suposición. Cuanto mayor es el tamaño de cada muestra menos estricta es la condición de normalidad y homocedasticidad (propiedad de igualdad de varianzas). Además, Anova es bastante robusto cuando el tamaño de cada población es equitativo.

3.1.1. Alternativas a Anova

Vamos a presentar alternativas para cuando no se cumplan alguna de las condiciones según el número de grupos.

- En el caso de que solamente tengamos dos muestras, se puede resumir como en la Figura 3.1. Si hay normalidad en las muestras utilizaríamos **t-test** cuando hay igualdad de varianzas y **test de Welch** cuando no tienen la misma varianza. Cuando las dos muestras no siguen una distribución normal optaríamos por equivalentemente **Mann-Whitney**, **test de rangos** o **Wilcoxon**.

$H_0: \mu_1 = \mu_0$ $H_1: \mu_1 \neq \mu_0$		NORMALIDAD	
		SI	NO
HOMOCEDASTICIDAD	SI	t-test	Mann-Whitney
	NO	Welch	Mann-Whitney

Figura 3.1: Tabla en la que se indican diferentes test para comparar la media de dos muestras según la normalidad y la homocedasticidad.

- Cuando tenemos n muestras con $n > 2$ pero no hay normalidad se utiliza el contraste de medianas $H_0 : Med_1 = \cdots = Med_n$, que es una versión no paramétrica del Anova, llamado **test Kruskal Wallis.** En resumen, tal y como podemos ver en la Figura 3.2

		NORMALIDAD	
		SI	NO
HOMOCEDASTICIDAD	SI	Anova	Kruskal Wallis
	NO	Welch	Kruskal Wallis

Figura 3.2: Tabla en la que se indican diferentes test para comparar la media de $n \geq 2$ muestras según la normalidad y la homocedasticidad.

Para resolver si las muestras cumplen las propiedades de normalidad y homocedasticidad se utilizan los siguientes test:

- Normalidad. Si la muestra es pequeña (< 30 elementos) se utiliza el **test Shapiro-Wilk**. En cambio, para muestras más grandes se usa el **test Kolmogorov-Smirnov.**
- Homocedasticidad. La **prueba de Levene** es la más utilizada para examinar esta característica. Tiene como hipótesis nula $H_0 : \sigma_1 = \cdots = \sigma_k$. Cuando las muestras son normales, se puede utilizar el **test Fisher** (para dos muestras, con $H_0 : \sigma_1/\sigma_2 = 1$) o el **test Bartlett** (para dos o más muestras de distinto tamaño).

3.2. Spherical k-means

El **Spherical k-means - SPKM** es un procedimiento de clasificación propuesto por Dhillon y Modha en 2001 de aprendizaje no supervisado. Actualmente, este método es de gran utilidad en la clasificación de documentos de texto donde la matriz de datos es *sparse*. Es necesario indicar el número de grupos o *clusters* que queremos. Es una variación del *k-mean* en la hiper-esfera (de cualquier dimensión) considerando *cosine dissimilarity* en vez de distancia Euclídea, cuyo objetivo es minimizar

$$\sum_i d(x_i, \pi_{c(i)}),$$

siendo

$$d(x_i, \pi_{c(i)}) = 1 - \cos(x_i, \pi_{c(i)}) = 1 - \frac{\langle x, \pi \rangle}{\|x\| \|\pi\|}$$

donde $X = \{x_1, \ldots, x_d\}$ es el conjunto de los vectores de $\mathbb{R}^w$ que queremos clasificar, k el número de *clusters*, $X = \{\pi_1, \ldots, \pi_k\}$ con $\pi_j \cap \pi_l = \varnothing$ cuando $j \leq l$ son las particiones y $c(i) \in \{1, \ldots, k\}$ indica las posibles agrupaciones.

Consideramos $\|x_i\|_2 = 1$, en otro caso tomar $\frac{x_i}{\|x_i\|_2}$. Para un *cluster* π denotamos

$$s(\pi) = \sum_{x \in \pi} x,$$

$$c(\pi) = \frac{s(\pi)}{\|s(\pi)\|}$$

y

$$q(\pi) = \sum_{x \in \pi} x^T c(\pi) = \|s(\pi)\|.$$

Por convenio se determina $q(\varnothing) = 0$ y por notación consideramos $c(\pi_j) = c_j$. Luego, la función objetivo de la que queremos maximizar es para k *clusters* es

$$Q(\{\pi_j\}_{j=1}^k) = \sum_{j=1}^k q(\pi_j) = \sum_{j=1}^k \sum_{x \in \pi_j} x^T c_j$$

Denotamos por $tol > 0$ la tolerancia del algoritmo a partir de la cual se detiene el proceso. Se establece la selección de la partición inicial para $t = 0$, que en general es aleatoria, $\{\pi_l^{(0)}\}_{l=1}^k$ que tiene asociados los vectores $c_1^{(0)}, \ldots, c_k^{(0)}$.

A continuación se calcula la siguiente posible partición. Primero se calcula

$$l^*(x) = \arg\max_j x^T c_j^{(t)},$$

es decir, se obtiene el índice de los vectores asociados a la partición que está más cerca considerando *cosine similarity*.

A partir de ahí, se construye la nueva partición

$$\pi_l^{(t+1)} = \{x \in X \mid l^*(x) = l\}, \quad 1 \leq l \leq k.$$

Ahora falta comprobar que esta nueva partición es "mejor"que la anterior bajo la tolerancia establecida. Para ello si

$$Q(\{\pi_j^{(t+1)}\}_{j=1}^k) - Q(\{\pi_j^t\}_{j=1}^k) > tol$$

repetir el proceso aumentando en uno el valor t, en otro caso obtendríamos la partición resultante del algoritmo.

Se trata de un método heurístico, que no proporciona siempre una solución óptima, en ocasiones proporciona un óptimo local. Encontrar el óptimo global es un problema mucho más complejo. Cuando tenemos gran cantidad de datos suele funcionar bien, pero cuando tenemos pocos da escasos resultados [20].

Los resultados obtenidos mediante *SPKM* dependen mucho de la elección inicial de los primeros k centros. Se puede encontrar en la literatura variaciones como **Spherical k-means++ - SPKM++** que aporta una estrategia en dicha elección sin realizar ninguna asunción sobre los datos, o incluso variaciones en las que intervienen *cadenas de Markov* [21].

La literatura acerca de este método es muy amplia y con especial aplicación a datos formados por palabras, destacar los aportada en [22].

Así como es fácil encontrar variaciones de este procedimientos, por ejemplo en [20] encontramos una variación en la que se alterna con una modificación que permite salirse del óptimo local.

3.3. Silhouette Plot

Vamos a utilizar **silhouette plot** para cuantificar cómo de buena es una clasificación. Esta forma de medir es independiente del algoritmo utilizado para la obtención de los *clusters*. Además también es independiente de las variables a las que representa cada grupo. Por esta razón, se suele utilizar para comparar distintos procedimientos con un mismo conjunto de datos, así como para comparar con distintos números de grupos puesto que en muchos casos es desconocido. Ambas aplicaciones se han realizado en [16].

Se basa en la disimilaridad de cada elemento con respecto a los distintos *clusters*. Para ello, es necesario que las distancias sean *ratio scale*, es decir, que una disimilaridad de 6 sea dos veces una de 3, como ocurre en la distancia Euclídea. Esto es conveniente de mencionar puesto que los *silhouette plots* son comunes en la clasificación de textos donde medir las disimilaridades no es trivial.

Vamos a proceder a realizar diferentes definiciones para introducirnos en el tema. Sea i un elemento cualquiera asignado al *cluster* A:

- $a(i) =$ media de las distancias con el resto de objetos de A.
- $d(i,C) =$ media de las distancias con todos los objetos de C, con $C \neq A$.
- $b(i) = \min_{C \neq A} d(i,C)$

El *cluster* B tal que alcanza la mínima distancia, en otras palabras, $d(i,B) = b(i)$, se denomina *vecino*, y sería la segunda opción si no pudiera asignarse i a A, ya que es el siguiente más cercano de media.

A partir de estas nociones se determina el valor $s(i)$ para cada elemento i como

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

Por convenio se establece $s(i) = 0$ cuando A contiene un único elemento y se asume que el número de *clusters* es mayor que 1. Por otro lado, es fácil ver que

$$-1 \leq s(i) \leq 1.$$

Estudiemos a continuación qué significado tienen los valores $s(i)$.

- Valores $s(i)$ próximos a 1 se tienen cuando el valor de $b(i)$ es mucho mayor a $a(i)$ y por tanto, el elemento i está en un *cluster* correcto, ya que la segunda opción, que sería B estaría bastante más lejos.
- Cuando $s(i)$ es casi 0, se tiene que $a(i) \approx b(i)$, luego, i estaría a la misma distancia media de A y B y no se podría decir que *cluster* es "mejor".
- Para valores de $s(i)$ cerca de -1 se tiene que $a(i)$ es mucho mayor que $b(i)$, por tanto se ha realizado una mala clasificación de i al *cluster* A y sería mejor asignarle B.

En conclusión, cuanto mayor sean los valores de $s(i)$ mejor es la clasificación y valores negativos de $s(i)$ implican una mala clasificación del elemento i.

Notar que en este análisis, para el caso de tener dos grupos, el hecho de cambiar un elemento de un *cluster* a otro implica modificar $s(i)$ a $-s(i)$.

Se puede realizar un estudio análogo en el que en vez de tratar con disimilaridades se utilicen similaridades con algunas variantes en las definiciones como por ejemplo tomar $b'(i) = \max_{C \neq A} d(i,C)$.

Una vez se ha calculado los valores $s(i)$ para todos los elementos i, se procede a dibujar los valores ordenados por *clusters* y de forma descendente. Es una manera gráfica muy intuitiva de ver el ajuste

de una clasificación, como podemos ver en la Figura 2.9(A-G). El hecho de que sean convexos indican categoriedad en los datos. Además falta de valores negativos y próximos a cero implican un buen ajuste.

Como hemos comentado anteriormente, se puede utilizar este algoritmo para comparar entre distintos métodos o entre distinta cantidad de *clusters*. Por ello, se puede calcular la media de todos los valores $s(i)$ y se asocia a cada clasificación y cuanto mayor sea ese valor, mejor es el ajuste.

Toda la información utilizada en esta sección se ha basado en [23].

3.4. Variable-centroid clustering

Se trata de un método de clasificación. Parte de un conjunto de variables previamente propuestas. Se establecen como centros de los *clusters* y se trata de asignar a cada elemento el *cluster* cuyo centro esta más cerca.

El problema de este método es que es necesario definir los centros, y por tanto conocer también el número de grupos. Una variante, cuando el número de variables es superior al de *clusters*, es elegir los centros aleatoriamente.

3.5. Mutual Information - MI

A continuación, presentamos un método y dos versiones del mismo que se encargan de medir la similitud entre dos particiones a través del concepto de *Entropía de Shannon*.

En numerosas áreas aparece la noción de entropía y en cada una de ellas, puede significar una cosa diferente. Por ejemplo, en física es un concepto de termodinámica que se refiere al desorden de un sistema o en informática es una magnitud de incertidumbre. En nuestro caso, la entropía indica cuánta información da un suceso. Cuanto mayor es la entropía, más capacidad de información, así como un evento que es poco probable proporciona más información. Aunque que tenga más capacidad no implica que nos proporcione más información. Podemos hacer un símil con respecto a un reloj. Si la aguja correspondiente a los segundo es regular no nos va a proporcionar información.

De una forma general, si consideramos una variable uniforme discreta con k sucesos con probabilidades p_k respectivas, la *entropía* del evento k se define como como

$$H(k) = -p_k \log_2 p_k.$$

Notar que $H(k) \geq 0$. Además, para una variable aleatoria X, la **Entropía de Shannon** es

$$H(X) = -\sum_{k \in X} p(k) \log_2 p(k).$$

La unidad de medida de la entropía son los bits. Este concepto es de gran utilidad en la Teoría de la Información. También se puede definir la *entropía condicionada* como

$$H(X|Y) = \sum_{y_i \in Y} p(y_i) H(X|y_i)$$

que trata de explicar la variabilidad de X que no depende de Y.

A partir de estas nociones, ya podemos introducir la **Mutual Information - MI**, que indica lo que aporta X a Y y se calcula mediante

$$MI(X,Y) = H(X) - H(X|Y).$$

Se trata de una medida que muestra la correlación no lineal de las dos variables. También tenemos que $MI(X,Y) \geq 0$ y si las variables son independientes entonces $MI(X,Y) = 0$.

Podemos ver gráficamente estos conceptos en la Figura 3.3, donde

$$I(X;X) = H(X)\,.$$

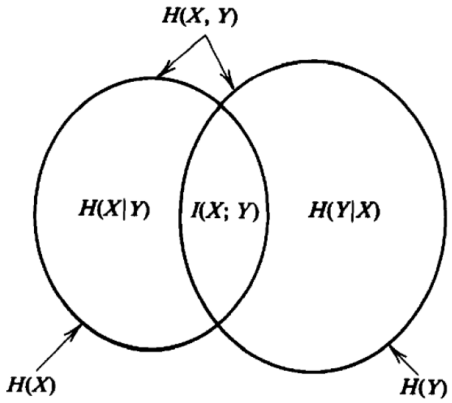

Figura 3.3: Representación gráfica de los concepto de *Entropía de Shanon* y *Mutual Information*. Imagen obtenida en [24].

3.5.1. Normalized Mutual Information - NMI

La *MI* es una correlación no lineal, no está escalada. Por eso se normaliza a través del máximo y así se puede comparar. La variación normalizada de *MI* es la **Normalized Mutual Information - NMI** que se define como

$$NMI(X,Y) = \frac{H(X) - H(X|Y)}{\text{máx}\{H(X), H(Y)\}} = \frac{MI(X,Y)}{\text{máx}\{H(X), H(Y)\}}.$$

Este valor pertenece al intervalo $[0,1]$, lo que nos va a permitir comparar, puesto que están en la misma escala. El inconveniente es que al normalizar ha perdido las unidades. Si es próxima a 1, entonces las variables explican lo mismo.

3.5.2. Adjust Mutual Information - AMI

Estos métodos tiene problemas porque estiman distribuciones. La **Adjust Mutual Information - AMI** se trata de una versión escalada de la versión normalizada *NMI* que corrige el sesgo positivo.

$$AMI(X,Y) = \frac{H(X) - H(X|Y) - E[H(X) - H(X|Y)]}{\text{máx}\{H(X), H(Y)\} - E[H(X) - H(X|Y)]} = \frac{MI(X,Y) - E[MI(X,Y)]}{\text{máx}\{H(X), H(Y)\} - E[MI(X,Y)]}$$

con $E[Z]$ denotando la esperanza de la variable Z.

En nuestro caso la vamos a aplicar a dos particiones del mismo espacio $U = \{U_1, \ldots, U_R\}$ y $V = \{V_1, \ldots, V_C\}$ para compararlas. Consideramos n_{ij} el número de elementos que están en la intersección de U_i y V_j, es decir $n_{ij} = \#(U_i \cap V_j)$. Definimos $a_i = \sum_{j=1}^{C} n_{ij}$ y $b_j = \sum_{i=1}^{R} n_{ij}$. De esta manera tenemos,

$$H(U) = -\sum_{i=1}^{R} \frac{a_i}{N} \log_2 \frac{a_i}{N}, \quad H(V) = -\sum_{j=1}^{C} \frac{b_i}{N} \log_2 \frac{b_i}{N}$$

Y por lo tanto,

$$MI(U,V) = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{n_{ij}}{N} \log_2 \frac{n_{ij}/N}{a_i b_j / N}.$$

Lo vamos a utilizar para detectar posibles relaciones lineales y no lineales entre variables (con un máximo de 5 variables). Se usa *AMI* como medida de similaridad.

En [16] se realiza un estudio heurístico en el que se dispone que poblaciones de neuronas sin estructura categórica, es decir, con datos uniformes se obtienen valores de *AMI* muy próximos a cero, mientras que para datos estructurados categoricamente se tiene como resultado más variación. Lo podemos ver en la Figura 3.4. Los resultados han sido obtenidos a través de datos sintéticos.

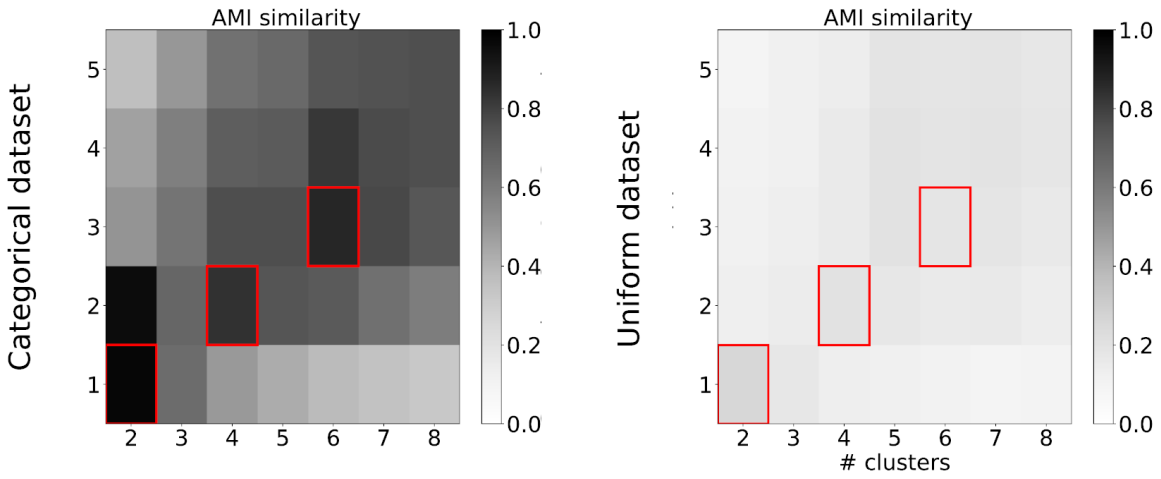

Figura 3.4: Valores de *AMI* dependiendo del número de *clusters* y variables para datos sintéticos categóricos en la imagen de la izquierda y sin estructura categórica en la imagen de la derecha. Imagen modificada de [16].

El mayor problema de la *MI* es el cálculo de las probabilidades condicionadas porque depende de la muestra que tengamos. Una propiedad de estos métodos es la simetría.

3.6. Jackknife

Jackknife es una técnica de remuestreo que se aplica al estimar un valor en un método para hacerlo más robusto, en otras palabras, para disminuir la varianza. Se utiliza para estimar el sesgo o la varianza de un estimador. El sesgo es la diferencia entre la esperanza de un estimador y el valor real del parámetro.

La manera de proceder es la siguiente: Para cada elemento de la muestra, se realiza la estimación con la submuestra que se obtiene eliminando esa observación de la muestra inicial. Finalmente, se realiza la media de todos los resultados.

Este método tiene n posibles submuestras, una por cada observación. Es el antecesor del **Boostrap**, lo que hace a *Jackknife* más rápido.

Dada una muestra de n registros, la técnica *Boostrap* consiste en la creación de B nuevas muestras de tamaño n a partir de la dada. Estas muestras se construyen eligiendo de manera aleatoria n elementos con repetición, en otras palabras, unos registros van a estar repetidos y otros se van a omitir.

En [16] se utiliza para estimar el error estándar de *AMI* mediante el estimador de *Jackknife* del error estándar:

$$std(AMI) = \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}(\theta_i - \theta_\cdot)^2}$$

donde θ_i es el valor estimado cuando quitamos el elemento i y $\theta_\cdot$ es la media de las n valores θ_i.

Capítulo 4

Aplicación

Para la realización de este trabajo se ha considerado oportuno realizar el procedimiento expuesto en el Capítulo 2, en lenguaje de programación R y la reproducción de sus mismos datos.

4.1. Código

Para comenzar, cargamos los paquetes necesarios, hemos indicado a la derecha con un comentario para que hemos utilizado cada paquete.

```
library(ggplot2)#Graficos
library(R.matlab)#Leer datos
library(text2vec)#Normalizar
library(skmeans)
library(cluster)#Pintar silhouette
library(lsa)#Distancia coseno
library(aricode)#AMI
```

Establecemos los directorios en los que tenemos los datos en las variables *PATH_RATES* y *PATH_TRIALS*.

Cargamos la media de las tasas de disparo, de manera que obtengamos una matriz $n \times m$ con n el número de neuronas y m la cantidad de *trial types*. Notad que hemos tenido que realizar un análisis previo en el que se estudia que tengan un comportamiento similar, es decir, consideramos aquí un único *pool*. El proceso se repite para cada *pool* y además también para cada ventana de tiempo

Vamos a tratar aquí únicamente con el *pool 1* y la ventana de tiempo *post-entrega*. Dependiendo cómo tengamos los datos, este paso puede verse modificado.

```
i_pool = 1
# Cargamos la media de las tasas de disparo
MAT_CONTENTS = readMat(con = PATH_RATES)
RATES = MAT_CONTENTS['Xall']
RATES2 = RATES[1][[1]]
MDATOS = RATES2[[i_pool]][[1]] # Cogemos el Pool 1
```

De esta manera, en *MDATOS* tenemos las tasas de disparo en el formato deseado. Ahora, vamos a centralizar y normalizar. En otras palabras, vamos a transformar los datos para que cada fila tenga media cero y norma 2 igual a 1.

Para ello utilizamos la función *normalize* del paquete *text2vec*.

```
CRATES = MDATOS - apply(X = MDATOS, MARGIN = 1, FUN = 'mean')
NORM_RATES = text2vec::normalize(CRATES, norm = "l2")
```

Una vez tenemos los datos situados en la hiper-esfera, añadimos las neuronas con comportamiento opuesto.

```
NORM_RATES = rbind(NORM_RATES, -NORM_RATES)
```

Nuestra matriz de datos normalizados *NORM_RATES* pasa a tener dimensión $2n \times m$.

A continuación, se carga la información respectiva a los *trial types* para proceder a calcular las variables.

Se trata de una matriz con $m \times 3$ con m el número de *trial types*. La primera columna se corresponde con la oferta de la Opción 1, en nuestro caso sería la cantidad de zumo A ofrecido. La segunda columna contiene la oferta de la Opción 2, es decir, la cantidad de zumo B ofrecido. Y la tercera columna la elección mayoritaria siendo 1 si es A y -1 si es el zumo B. Además también cargamos los valores relativos. Para cada *pool* vamos a tener solamente uno.

```
MAT_CONTENTS_TRIAL = readMat(con = PATH_TRIALS)
TRIAL_TYPES_TOTAL = MAT_CONTENTS_TRIAL[['trialtypes']]

TRIAL_TYPES = TRIAL_TYPES_TOTAL[[i_pool]][[1]] #Consideramos solo el Pool 1
RELATIVE_VALUES = MAT_CONTENTS_TRIAL[['relvals']]
```

Definimos las variables que vamos a considerar como centros cuando apliquemos *variable-centroid clustering*. Aquí podemos ver la importancia de considerar la hiper-esfera, puesto que sino fuera así, no podríamos comparar los datos con las variables. Inicialmente se han considerando 10 variables, pero después veremos se decide colapsar 4 de ellas, ya que ofrecen información complementaria.

```
#Definimos las variables
VARIABLE_NAMES = c('chosen value', 'other value', '(chosen-other) value',

                   '(other/chosen) value', 'chosen number', 'offer value A',

                   'offer value B', 'chosen juice', 'chosen value A', 'chosen value B')
```

Calculamos los datos referentes a las variables con la información de los *trial types* y el valor relativo. Este apartado dependerá de la tarea que ha sido realizada.

```
VARIABLE_VALUES = matrix(0L, nrow =length(VARIABLE_NAMES), ncol = dim(TRIAL_TYPES)[1])
for (i_tt in 1:dim(TRIAL_TYPES)[1]) {
  tt = TRIAL_TYPES[i_tt,]
  numbers = tt[1:2]
  if (tt[3] == 1) {
    chosen_index = 1
    other_index = 2
  } else {
    chosen_index = 2
    other_index = 1
  }
  values = numbers
  values[1] = values[1]*RELATIVE_VALUES[i_pool]
  VARIABLE_VALUES[match('chosen value',VARIABLE_NAMES),i_tt] = values[chosen_index]
  VARIABLE_VALUES[match('other value',VARIABLE_NAMES),i_tt] = values[other_index]
  VARIABLE_VALUES[match('(chosen-other) value',VARIABLE_NAMES),i_tt] = values[chosen_index] -
      values[other_index]
  VARIABLE_VALUES[match('(other/chosen) value',VARIABLE_NAMES),i_tt] = values[other_index]/
      values[chosen_index]
  VARIABLE_VALUES[match('chosen number',VARIABLE_NAMES),i_tt] = #numbers[chosen_index]
    values[chosen_index]
  VARIABLE_VALUES[match('offer value A',VARIABLE_NAMES),i_tt] = values[1]
  VARIABLE_VALUES[match('offer value B',VARIABLE_NAMES),i_tt] = values[2]
  VARIABLE_VALUES[match('chosen juice',VARIABLE_NAMES),i_tt] = other_index - 1#Para que sea 1A
      /0B

  if (chosen_index == 1) {
    VARIABLE_VALUES[match('chosen value A',VARIABLE_NAMES),i_tt] = values[chosen_index]
    VARIABLE_VALUES[match('chosen value B',VARIABLE_NAMES),i_tt] = 0
  } else {
    VARIABLE_VALUES[match('chosen value A',VARIABLE_NAMES),i_tt] = 0
```

```
    VARIABLE_VALUES[match('chosen value B',VARIABLE_NAMES),i_tt] = values[chosen_index]
  }
}
```

Normalizamos la matriz correspondiente a las variables.

```
NORM_VARIABLES = text2vec::normalize(VARIABLE_VALUES, norm = "l2")
```

Colapsamos las variables oferta de A con oferta de B y valor de A escogido con valor de B escogido. De esta manera, cuando después calculemos las posibles combinaciones, no aparecerán juntas.

```
TO_COLLAPSE = rbind(c('offer value A', 'offer value B'),
                    c('chosen value A', 'chosen value B'))

INDICES_COLLAPSE = matrix(0L, nrow = dim(TO_COLLAPSE)[1], ncol = dim(TO_COLLAPSE)[2])

for (i in 1:dim(TO_COLLAPSE)[1]) {
    for (j in 1:dim(TO_COLLAPSE)[2]){
      INDICES_COLLAPSE[i,j] = match(c(TO_COLLAPSE[i,j]),VARIABLE_NAMES)
    }
}
```

Establecemos un número máximo de variables a considerar. Se recomienda no utilizar un número muy alto puesto que el coste computacional aumenta notablemente, hemos considerado 5 un número razonable. También declaramos el rango de *clusters* que queremos utilizar para el *spherical k-means*. Hemos determinado de 2 a 10 *clusters*.

```
# Numero maximo de variables a considerar en variable-centroid clustering
MAX_N_VARIABLES = 5
# Numero maximo de cluster para spherical k-means
RANGE_N_CLUSTERS = seq(from = 2, to = 10)
```

Definimos unas funciones previamente para que el código sea más intuitivo. La función *calcula_comb* quita de todas las posibles combinaciones aquellas indicadas en *indice_collapse*.

```
calcula_comb <- function (comb, indices_collapse) {
  if (!is.null(indices_collapse)) {
      valid = rep(x=TRUE, times = dim(comb)[1])
      for (collapse in 1:dim(indices_collapse)[1]){#Para cada fila
        # Count number of index occurrences
        occurrences = rep(x = 0,times = dim(comb)[1])
        for (c_index in 1:dim(indices_collapse)[2]) {
          occurrences = occurrences + rowSums(comb == indices_collapse[collapse, c_index])
        }
        valid = valid&(occurrences == 0 | occurrences == dim(indices_collapse)[2])
      }
      comb = comb[valid,]
    }
  return(comb)
}
```

También definimos una función que calcula las particiones de *variable-centroid clustering* dados los datos normalizados y los centros a través de la función *cosine* del paquete *lsa*.

```
centroid_cosine_cluster <- function (norm_rates, centros) {
  labels = c()
    for (i in 1:dim(norm_rates)[1]) {
      kernel = cosine(cbind(as.matrix(norm_rates[i,]), t(centros)))[1,-1]
      labels = cbind(labels, which.max(kernel))#Equivale a argmax
    }
  return(labels)
}
```

Aplicamos el resto del algoritmo diferenciando cuando consideramos una única variable puesto que se trata como vector y no como matriz y la manera de proceder difiere levemente. Notar que la función *AMI* pertenece al paquete *aricode*.

```
#Inicializamos
info = matrix(0L, nrow = MAX_N_VARIABLES, ncol = length(RANGE_N_CLUSTERS))
sel_var_ind = matrix(0L, nrow = MAX_N_VARIABLES, ncol = length(RANGE_N_CLUSTERS))

for (i_n_clusters in 1:length(RANGE_N_CLUSTERS)) {
  Y_CLUST = list()
  alg = skmeans(x = NORM_RATES, k = RANGE_N_CLUSTERS[i_n_clusters])
  Y_CLUST = alg$cluster
  for (i_n_var in 1:MAX_N_VARIABLES) {
    var_combs = t(combn(dim(NORM_VARIABLES)[1], i_n_var))#Calculamos todas posibles
        combinaciones
    var_combs = calcula_comb(var_combs, INDICES_COLLAPSE)#Quitamos las indicadas en INDICES_
        COLLAPSE
    cand = c()
    if (is.vector(var_combs)) {
      for (i_v_c in 1:length(var_combs)) {
        cand_pool = c()
        centroids = rbind(NORM_VARIABLES[var_combs[i_v_c], ], -NORM_VARIABLES[var_combs[i_v_c
            ], ])
        y_var = centroid_cosine_cluster(NORM_RATES, centroids)
        cand_pool = cbind(cand_pool, AMI(as.vector(Y_CLUST), as.vector(y_var)))
        cand = cbind(cand, sum(cand_pool))
      }
    } else {
      for (i_v_c in 1:dim(var_combs)[1]) {
        cand_pool = c()
        centroids = rbind(NORM_VARIABLES[var_combs[i_v_c,], ], -NORM_VARIABLES[var_combs[i_v_c
            ,], ])
        y_var = centroid_cosine_cluster(NORM_RATES, centroids)
        cand_pool = cbind(cand_pool, AMI(as.vector(Y_CLUST), as.vector(y_var)))
        cand = cbind(cand, sum(cand_pool))
      }
    }
    sel_var_ind[i_n_var,i_n_clusters] = var_combs[which.max(cand)]
    info[i_n_var, i_n_clusters] = max(cand)
  }
}
```

Finalmente, hemos obtenido una matriz con los distintos valores de *AMI* para el número máximo de variables indicado, así como para el rango de *clusters*. También una variable donde hemos añadido las variables a las que corresponde el máximo valor de *AMI*.

4.1.1. Representación de los resultados

Se considera útil la reproducción de los resultados para ir comprobando y detectar posibles errores. Se presentan distintas opciones posibles. Algunas de ellas han sido utilizadas en el Capítulo 2.

Para la representación de los *silhouette plots* es necesario volver a calcular las particiones puesto que no hemos guardado dicha información. Con un máximo de 8 valores distintos del número de *clusters* a considerar se puede realizar de siguiente manera.

```
c8 = c("tomato","orange","yellow", "olivedrab2","forest green","aquamarine2", "royalblue", "
    purple2", "goldenrod4", "gray20")
RANGE_N_CLUSTERS = seq(from = 2, to = 10)
Y_CLUST = list()
Y_CLUST_CENTERS = list()
par(mfrow = c(1, length(RANGE_N_CLUSTERS)))
for (i_n_clusters in 1:length(RANGE_N_CLUSTERS)) {
  alg = skmeans(x = NORM_RATES, k = RANGE_N_CLUSTERS[i_n_clusters])
  Y_CLUST[[i_n_clusters]] = alg$cluster
  Y_CLUST_CENTERS[[i_n_clusters]] = alg$prototypes
  sil = silhouette(alg)
  plot(sil, main = paste("k = ", RANGE_N_CLUSTERS[i_n_clusters]),
```

```
      col = c8[1:RANGE_N_CLUSTERS[i_n_clusters]],
      do.n.k = FALSE, do.clus.stat = FALSE, xlab="", sub = paste("Media de s(i) = ",format(
          round(mean(sil[,3]), 4), nsmall = 2) ))
}
```

Para ver graficamente los resultados de la matriz *AMI* procedemos mediante el paquete *plot.matrix*.

```
library(plot.matrix)#plot AMI
library(viridis)#colores
colnames(info) = RANGE_N_CLUSTERS
plot(info, fmt.cell = '%.2f', fmt.key = "%.2f", key = list( cex.axis = 0.65), breaks = 15,
    xlab = "Numero de clusters", ylab =  "Numero variables", main = "AMI", col = magma)
# Colores  'viridis' 'magma', 'plasma', 'inferno', and 'cividis'
```

4.2. Datos

Figshare es un repositorio libre de datos, donde puedes encontrar pasos intermedios y resultados de multitud de artículos científicos.

Tanto la base de datos como el código en Python en el que se ha basado este algoritmo, son de libre acceso y han sido consultados en Figshare en:

`https://doi.org/10.6084/m9.figshare.9844349.v1`

Hemos considerado como datos base, los archivos con extensión .mat y a partir de ahí hemos desarrollado la adaptación del código. En la sección anterior, se explica detalladamente cómo hemos utilizado dichos datos y cómo extraer la información en el programa informático R.

Además también se puede encontrar archivos .npz con extensión relativa a Python donde aparecen los pasos intermedios, por ejemplo los valores normalizados de las variables. Con estos datos hemos ido comparando con los obtenidos mediante R y hemos visto consistencia en los resultados.

4.3. Versiones

Para la reproducción del procedimiento se han utilizado la versión 3.6.1 de R. Los paquetes usados tienen la versión indicada en la Figura 4.1.

Paquete	Versión
ggplot2	3.2.1
R.matlab	3.6.2
text2vec	0.6
skmeans	0.2-11
cluster	2.1.0
lsa	0.73.2
aricode	1.0.0
plot.matrix	1.4
viridis	0.5.1

Figura 4.1: Versiones de los paquetes utilizados en R.

Bibliografía

[1] CONTEXTO HISTÓRICO, *Neurociencia*, https://www.medicalnewstoday.com/articles/248680#history.

[2] CONTEXTO HISTÓRICO, *Neurociencia - Velocidad*, http://book.bionumbers.org/how-fast-are-electrical-signals-propagated-in-cells/.

[3] IMAGEN NEURONA RAMÓN Y CAJAL, *Imagen obtenida en:* https://www.nytimes.com/es/2017/02/21/espanol/cultura/santiago-ramon-y-cajal-el-hombre-que-dibujo-los-secretos-del-cerebro.html.

[4] IMAGEN SNC Y SNP, *Imagen modificada de:* , https://www.news-medical.net/health/What-is-the-Nervous-System-(Spanish).aspx.

[5] IMAGEN PARTES NEURONA, *Imagen obtenida en:* https://milrespuestas5.blogspot.com/2016/09/que-relacion-hay-entre-ramon-y-cajal-y.html.

[6] NEURONAS Y SPIKES, *Información acerca de neuronas y spikes:* , https://icwww.epfl.ch/~gerstner/SPNM/node3.html.

[7] FISIOLOGÍA HUMANA, *Stuart Ira Fox, 13ª Ed. ISBN: 978-607-15-1151-5.*

[8] NEURAL CORRELATES OF A POSTPONED DECISION REPORT, *Luis Lemus, Adrián Hernández, Rogelio Luna, Antonio Zainos, Verónica Nácher y Ranulfo Romo. PNAS Octubre 2007, vol. 104, num 43 17174–17179.*

[9] THE CAPACITIES OF HUMANS AND MONKEYS TO DISCRIMINATE BETWEEN VIBRATORY STIMULI OF DIFFERENT FREQUENCY AND AMPLITUDE: A CORRELATION BETWEEN NEURAL EVENTS AND PSYCHOLOGICAL MEASUREMENTS, *LaMotte RH, Mountcastle VB (1975) J Neurophysiol 38:539–559.*

[10] DECODING A PERCEPTUAL DECISION PROCESS ACROSS CORTEX, *A. Hernández, V. Nácher, R. Luna, A. Zainos, L. Lemus, M. Alvarez, Y. Vázquez, L. Camarillo, y R. Romo, 2010.*

[11] WHAT THE ORBITOFRONTAL CORTEX DOES NOT DO, *Thomas A Stalnaker, Nisha K Cooch, and Geoffrey Schoenbaum*, NatNeurosci.2015May;18(5):620\T1\textendash627. doi:10.1038/nn.3982.

[12] DISSOCIABLE COMPONENTS OF RULE-GUIDED BEHAVIOR DEPEND ON DISTINCT MEDIAL AND PREFRONTAL REGIONS, *Buckley, Mansouri, Hoda, Mahboubi, Browning, Kwok Phillips, Tanaka, 2009. Science (New York, N.Y.). 325. 52-8. 10.1126/science.1172377.*

[13] ORBITOFRONTAL CORTEX ENCODES WILLINGNESS TO PAY IN EVERYDAY ECONOMIC TRANSACTIONS, *Hilke Plassmann, John O'Doherty and Antonio Rangel Journal of Neuroscience 12 September 2007, 27 (37) 9984-9988; DOI: https://doi.org/10.1523/JNEUROSCI.2131-07.2007.*

[14] CONTEXT-DEPENDENT COMPUTATION BY RECURRENT DYNAMICS IN PREFRONTAL CORTEX, *Mante V, Sussillo D, Shenoy KV, Newsome WT* , `Nature.2013;503(7474):78\T1\ textendash84.pmid:24201281`.

[15] SPATIAL TRANSFORMATIONS IN THE PARIETAL CORTEX USING BASIS FUNCTIONS, *Pouget A, Sejnowski TJ*.

[16] CATEGORICAL ENCODING OF DECISION VARIABLES IN ORBITOFRONTAL CORTEX, *Arno Onken, Jue Xie, Stefano Panzeri, Camillo Padoa-Schioppa, 2019*.

[17] NEURONAL ORIGINS OF CHOICE VARIABILITY IN ECONOMIC DECISIONS, *Camillo Padoa-Schioppa, Neuron 80, 1322–1336, December, 2013*.

[18] NEURONS IN ORBITOFRONTAL CORTEX ENCODE ECONOMIC VALUE, *Padoa-Schioppa C, Assad JA. Nature. 2006;441(7090):223–6*.

[19] NEURONAL REMAPPING AND CIRCUIT PERSISTENCE IN ECONOMIC DECISIONS, *Xie J, Padoa-Schioppa C. Nat Neurosci. 2016;19(6):855–61*.

[20] ITERATIVE CLUSTERING OF HIGH DIMENSIONAL TEXT DATA AUGMENTED BY LOCAL SEARCH, *Dhillon IS, Guan Y, Kogan J, 2002,In Proceedings of the Second IEEE International Conference on Data Mining, pp. 131–138. Maebishi, Japan*.

[21] A FASTER SAMPLING ALGORITHM FOR SPHERICAL K-MEANS, *Rameshwar Pratap, Anup Deshmukh, Pratheeksha Nair, Tarun Dutt, Proceedings of Machine Learning Research 95:343-358, 2018*.

[22] CONCEPT DECOMPOSITIONS FOR LARGE SPARSE TEXT DATA USING CLUSTERING, *INDERJIT S. DHILLON, DHARMENDRA S. MODHA, Machine Learning, 42, 143–175, 2001*.

[23] SILHOUETTES: A GRAPHICAL AID TO THE INTERPRETATION AND VALIDATION OF CLUSTER ANALYSIS, *Peter J.Rousseeuw, Journal of Computational and Applied Mathematics 20 (1987) 53-65*.

[24] STATISTICAL ANALYSIS OF NEURAL CORRELATES IN DECISION-MAKING, *Marina Martinez-Garcia, TESI DOCTORAL UPF / 2014*.