**Autenticación de usuarios basada en reconocimiento de patrones EEG**

El proceso de experimentación implicó la recopilación de un conjunto de bases de datos denominado ‘DataSet IIb’, ofrecido públicamente por el Instituto Fraunhofer para los Sistemas de Comunicación Abiertos, el *Intelligent Data Analysis Group* (Klaus-Robert Müller, Benjamin Blankertz), el campus Benjamin Franklin de la Universidad de Medicina de Berlin y el *Neurophysics Group* Gabriel Curio para la *BCI competition IV*. Cada elemento de la antedicha colección no es sino una serie de valores enteros de 16 bits que representan el voltaje encefálico generado por nueve sujetos durante 159 experimentos[[1]](#footnote-1), cada uno con duración fija de dos segundos.

En dichas pruebas, se solicitó a los participantes imaginar el movimiento, ya sea de la mano izquierda o derecha, siendo evidente la presencia de dos clases de tarea mental a las que se hará referencia en el presente documento como y . Continuando con la descripción de los datos disponibles, es menester indicar que, la recopilación de los valores antedichos se realizó a una frecuencia de muestreo de 250 Hz, empleando los canales 2, 3 y 4 de la zona centro de la corteza cerebral, misma que se etiqueta bajo la leyenda “C” de acuerdo con el estándar internacional 10-20. Todo lo anterior, se organiza en matrices de la forma con los índices representando el canal, número de muestra en el tiempo y número de experimento respectivamente.

La prueba central de este documento, desarrollada en el lenguaje de programación Python 3.7.3, consiste en la elaboración de un clasificador bayesiano capaz de distinguir entre el movimiento del sujeto y , así como el movimiento del sujeto y . Expresando lo anterior en términos menos ortodoxos, sería adecuado afirmar que, el algoritmo fabricado debe ser capaz de identificar al responsable de cada movimiento. Por practicidad se optó por realizar únicamente comparaciones binarias entre sujetos, aunque se pretende que en productos posteriores el clasificador sea capaz de reconocer la autoría de un patrón neuronal entre los nueve posibles sujetos.

Pasando a detalles de la implementación, es importante destacar que, dado el hecho de que los datos citados inicialmente se distribuyen exclusivamente en el formato de compatibilidad para el ambiente MATLAB “.mat” se requirió el uso del módulo *scipy.io*, que permitió su importación y subsecuente transformación en una lista de listas de Python. Adicionalmente se empleó *numpy* para la operación de matrices y la extracción de características como varianza, media y desviación estándar; *scypi.signal* para el filtrado y remuestreo y *scipy.stats.norm* para el cálculo de la funcion de densidad de probabilidad gaussiana

De forma general, el programa sigue un modelo de inteligencia artificial supervisado, requiriendo inherentemente de dos fases: una de entrenamiento con datos previa y debidamente etiquetados, a los cuales se ha hecho alusión previamente, y una de clasificación, en la que se introducen valores desconocidos para probar la efectividad del adiestramiento. Así pues, el software elaborado recibe como parámetros los números de sujeto con los que se pretende realizar dicha etapa de aprendizaje, así como los detalles con que se llevará a cabo tal procedimiento, tales como: la cantidad de experimentos a considerar[[2]](#footnote-2), la constante denotando la frecuencia de muestreo, las bandas de frecuencia y que entre las cuales se filtrarán los datos, la clase por la que se pretende distinguir a ambos sujetos y los canales de recopilación que se deben considerar.

Subsecuentemente, el algoritmo se encarga de llevar a la memoria principal las bases de datos de y obteniendo, de acuerdo con lo estipulado inicialmente, dos listas de listas o matrices de la forma y , las cuales poseen un formato inteligible por Python. En ellas, se procede a aplicar un filtro Butterworth paso banda, que emplea empíricamente el coeficiente 4 y las bandas de frecuencia y proporcionadas. Después de ello y, por cuestiones de seguridad, el software calcula el valor mínimo entre el número de experimentos de , y la variable . De lo contrario, es posible solicitar un entrenamiento con más datos de los disponibles en las colecciones. Dicho valor sobrescribe a , perdiendo su magnitud inicial.

Subsecuentemente, el algoritmo realiza la extracción de características, calculando la varianza de todos los elementos de y Los resultados se almacenan en dos nuevas matrices independientes y cuyas dimensiones son , donde representa el número de elementos en y el número de elementos en . Por cuestiones de practicidad, se indicarán las dimensiones antedichas, simplemente como puesto que el valor uno es irrelevante.

Una vez adquiridas las matrices anteriores y y, con el objetivo de concluir el proceso de entrenamiento, se procede a calcular la media y desviación estándar . Dicha labor, deriva en la creación de cuatro vectores , , y que representan la media y desviación estándar del contenido de cada canal de y . Como es de esperar, dichos vectores poseen el mismo número de elementos que , ya que cada uno representa una medida estadística de cada canal en . Con esto, se concluye el proceso de entrenamiento del algoritmo, iniciando la fase de prueba.

Dicha fase, consiste en tomar cada experimento número de la matriz y calcular la probabilidades de que dicho experimento pertenezca a y . Lo cual se realiza siguiendo la siguiente ecuación, que no es sino la aplicación de la funcion de probabilidad gaussiana dado

Fórmula 1, 2. En la ecuación a la izquierda, se calculan las probabilidades de que, en el canal , el experimento número , guardado en la matriz , pertenezca al . Lo anterior, usando la función de densidad probabilística de Gauss con media y desviación estándar dada por el valor de los vectores y respectivamente. Como es de esperarse, se generarán tantas probabilidades como elementos haya en , es decir, la cardinalidad de [[3]](#footnote-4). Todas estas, se multiplican, siguiendo la hipótesis de variables independientes de Bayes para obtener un valor único de probabilidad de pertenencia al sujeto El proceso, se repite, usando ahora las medias y desviaciones estándar características del . Finalmente, el clasificador compara ambas probabilidades y asigna el experimento a la clase que presente una probabilidad superior. En este punto, verificar si el clasificador ha acertado es trivial. Puesto que, se le otorgó al clasificador un valor que de antemano se sabía pertenecía al sólo basta compararlo con el resultado que ha presentado el clasificador. Si este coincidió en que el valor pertenecía a , entonces se dice que ha acertado, fracasando en caso contrario. Al final, es posible tener un total de aciertos para el , a los que se denominarán bajo el nombre

Subsecuentemente, el experimento se repite ahora empleando los experimentos que de antemano se sabe pertenecen al , así pues, el cálculo de la probabilidad se realiza de acuerdo con las siguientes ecuaciones

Como es posible observar el procedimiento es idéntico, excepto que ahora se ha calculado la probabilidad, dados valores de la matriz y no como previamente se había hecho. En este caso, el clasificador nuevamente emitirá su juicio, sin embargo, en este caso, inherentemente, se contará como acierto si el clasificador indica que la tarea mental fue realizada por el . A este total de aciertos se le denominará bajo el nombre Una vez calculados los totales de aciertos, se calcula la efectividad del algoritmo dada por .

Una vez fabricado el algoritmo y, puesto que una efectividad alta tras una única ejecución podría ser simplemente producto del azar, se decidió repetir el experimento en todas las posibles combinaciones y usando y usando también la clase de movimiento 2. Los resultados obtenidos se plasman en la siguiente tabla

1. Las bases de datos originales mostraban variabilidad en el número de experimentos. Para evitar que dicha inequidad afectase el proceso de clasificación, se redujeron todas las colecciones a 159 experimentos. [↑](#footnote-ref-1)
2. Es menester introducir esta variable para evitar el sobre entrenamiento del clasificador. Posteriormente en el documento se exhibirá con mayor detalle dicha problemática [↑](#footnote-ref-2)
3. Como es posible observar, en la función no se itera usando como índice el valor especificado en el arreglo . En lugar de ello se emplean los números naturales desde 1 hasta la cardinalidad de . Esto se hace, puesto que, en la matriz representa el número de canal entre los seleccionados por el usuario, mientras que el valor en representa el número de canal entre todos los disponibles. Si, por ejemplo, en el arreglo se indica usar el canal 3 únicamente y asumiendo el mismo número de experimentos que en la prueba mostrada en la parte superior, tendría las dimensiones . Si se intentara acceder al canal con índice 3, se saldría del arreglo. [↑](#footnote-ref-4)