

**DPTO. DE INGENIERÍA ELECTRÓNICA
ÁREA DE TEORÍA DE LA SEÑAL Y COMUNICACIONES
ESCUELA SUPERIOR DE INGENIEROS
INGENIERO DE TELECOMUNICACIÓN
UNIVERSIDAD DE SEVILLA**

PROYECTO FIN DE CARRERA

**APLICACIÓN DE LOS MÉTODOS DE
SEPARACIÓN CIEGA DE SEÑALES COMO
AYUDA A LA MEJORA DE SEÑALES
BIOMÉDICAS**

**AUTOR: IVÁN DURÁN DÍAZ
TUTOR: DR. SERGIO ANTONIO CRUCES ÁLVAREZ**

JULIO DE 2001

PROYECTO FIN DE CARRERA

TÍTULO: Aplicación de los métodos de Separación Ciega de Fuentes como ayuda a la mejora de señales biomédicas

TRIBUNAL CALIFICADOR:

PRESIDENTE:

VOCAL:

SECRETARIO:

CALIFICACIÓN:

FECHA DE LECTURA:

A mi padre, que no pudo verlo.

A mi madre y mi hermano.

RESUMEN

En medicina existen medidas que se toman a partir del cuerpo humano con múltiples sensores simultáneamente. Esto conduce a que en cada sensor aparezcan mezcladas señales procedentes de varias fuentes independientes. surgen así dos problemas fundamentales: por un lado la aparición de artefactos (señales procedentes de fuentes que no se deseaba estudiar y que se encuentran mezcladas con las señales de interés); por otra parte, las propias señales de interés están mezcladas también, por lo que su estudio individualizado se hace muy difícil o imposible.

Es, pues, necesario un método que permita la separación de todas las fuentes que aparecen mezcladas en los sensores. Ese método es la denominada Separación Ciega de Fuentes (BSS: Blind Source Separation), que consigue separar un número de fuentes igual al de sensores (es decir, debe haber tantos o más sensores que fuentes) desconociendo las propiedades estadísticas de dichas fuentes. Un método derivado es la Extracción Ciega de Fuentes (BSE: Blind Source Extraction), que únicamente separa (extrae) un conjunto de las fuentes presentes en las observaciones (los datos). Éste método, combinado con la deflación (que consiste en eliminar de los datos las fuentes extraídas), se convierte en un método alternativo para llevar a cabo la separación de todas las fuentes.

En este proyecto se estudia la aplicación de la BSS, o del Análisis de Componentes Independientes (ICA: Independent Component Analysis) a las señales tomadas en medidas de medicina para la solución de los dos problemas que se han enunciado.

Se hará asimismo un estudio acerca del funcionamiento de varios algoritmos existentes para la separación y la extracción de fuentes aplicándolos a unos datos concretos (los correspondientes a un magnetoencefalograma o MEG de 122 sensores y 17730 muestras) para extraer y eliminar los artefactos existentes en dichos datos.

ÍNDICE

| | |
|--|----|
| CAPÍTULO 1..... | 1 |
| 1.1 BSS Y BSE | 2 |
| 1.2 SEÑALES BIOLÓGICAS | 5 |
| 1.3 ALGORITMOS Y SOFTWARE..... | 6 |
| 1.4 ESTRUCTURA DEL PROYECTO | 6 |
| CAPÍTULO 2..... | 9 |
| 2.1 MEZCLAS INSTANTÁNEAS | 10 |
| 2.1.1 Caso sin ruido..... | 10 |
| 2.1.2 Caso con ruido | 11 |
| 2.1.3 Separabilidad e identificabilidad..... | 12 |
| 2.1.4 Independencia y HOS | 14 |
| 2.1.5 ICA..... | 17 |
| 2.2 CRITERIOS DE SEPARACIÓN | 18 |
| 2.2.1 Funciones contraste | 18 |
| 2.2.2 Mínima información mutua | 21 |
| 2.2.3 Máxima verosimilitud..... | 22 |
| 2.2.4 Principios INFOMAX y ENTMAX..... | 24 |
| 2.2.5 Relaciones entre criterios | 26 |
| 2.2.6 Desajuste entre el modelo y la realidad..... | 26 |
| 2.3 EXTRACCIÓN CIEGA DE FUENTES..... | 27 |
| 2.3.1 Deflación..... | 28 |
| 2.4 CONCLUSIONES | 28 |
| CAPÍTULO 3..... | 31 |
| 3.1 APLICACIÓN DEL ICA AL ECG..... | 32 |
| 3.1.1 Extracción del ECG del feto | 33 |
| 3.1.2 BSS de parámetros del ECG..... | 35 |

| | | |
|-----------------|--|----|
| 3.2 | ICA APLICADO A fMRI..... | 37 |
| 3.3 | APLICACIÓN DE ICA A EEG Y MEG..... | 39 |
| 3.3.1 | MEG..... | 44 |
| 3.3.2 | Eliminación de artefactos..... | 45 |
| 3.3.3 | ERP | 46 |
| 3.3.4 | Algunas aplicaciones concretas aún no resueltas..... | 48 |
| 3.4 | CONCLUSIONES | 50 |
| CAPÍTULO 4..... | | 53 |
| 4.1 | ALGORITMO DE EXTRACCIÓN EN PRESENCIA DE RUIDO GAUSSIANO..... | 53 |
| 4.1.1 | Algoritmo CII de partida..... | 54 |
| 4.1.2 | Algoritmo de BSE a partir del algoritmo CII..... | 55 |
| 4.2 | ALGORITMO DE BSE DE FUENTES DE INTERÉS | 57 |
| 4.3 | ALGORITMO FAST-ICA..... | 59 |
| 4.3.1 | Relación con el algoritmo del apartado 4.1. | 60 |
| 4.4 | ALGORITMO INFOMAX | 61 |
| 4.5 | CONCLUSIONES | 65 |
| CAPÍTULO 5..... | | 67 |
| 5.1 | EXTRACT_QN | 68 |
| 5.1.1 | bsseQN..... | 68 |
| 5.1.2 | extract_QN..... | 68 |
| 5.2 | EXTRACT2..... | 70 |
| 5.3 | FASTICA..... | 70 |
| 5.3.1 | pow3..... | 73 |
| 5.3.2 | tanh..... | 74 |
| 5.3.3 | gauss..... | 74 |
| 5.3.4 | skew | 74 |
| 5.4 | MATLAB ICA/EEG..... | 74 |
| 5.4.1 | runica..... | 75 |
| 5.4.2 | Pequeños programas para la utilización del toolbox..... | 76 |
| 5.4.3 | Archivo de sensores 2-D | 90 |
| 5.5 | CONCLUSIONES | 91 |
| CAPÍTULO 6..... | | 93 |
| 6.1 | OBSERVACIONES | 93 |
| 6.2 | COMPARACIÓN DE ALGORITMOS | 96 |

| | | |
|--|--|-----|
| 6.2.1 | runica I | 97 |
| 6.2.2 | runica II..... | 99 |
| 6.2.3 | fastica I..... | 101 |
| 6.2.4 | fastica II..... | 102 |
| 6.2.5 | fastica III..... | 103 |
| 6.2.6 | fastica IV..... | 104 |
| 6.2.7 | fastica V | 105 |
| 6.2.8 | extract_QN..... | 107 |
| 6.2.9 | extract2..... | 108 |
| 6.2.10 | runica III..... | 109 |
| 6.2.11 | runica IV | 110 |
| 6.2.12 | fastica VI..... | 112 |
| 6.2.13 | fastica VII..... | 114 |
| 6.2.14 | fastica VIII | 116 |
| 6.2.15 | fastica IX..... | 117 |
| 6.2.16 | extract_QN II | 118 |
| 6.2.17 | extract2 II | 120 |
| 6.2.18 | Comparación de las extracciones totales | 121 |
| 6.2.19 | Comparación de las extracciones y separaciones con reducción de componentes principales..... | 123 |
| 6.3 | SEÑALES LIBRES DE ARTEFACTOS | 125 |
| CAPÍTULO 7..... | | 127 |
| APÉNDICE. NOTACIÓN Y ABREVIATURAS..... | | 131 |
| NOTACIÓN..... | | 131 |
| ABREVIATURAS | | 133 |
| REFERENCIAS | | 135 |

CAPÍTULO 1

INTRODUCCIÓN

En muchas ocasiones se toman medidas de una magnitud (potencial eléctrico, flujo magnético, etc.) simultáneamente con varios sensores distribuidos en distintos puntos para obtener señales emitidas por diversas fuentes. Se pretende recoger en sensores próximos a cada una de esas fuentes las señales emitidas por éstas. Esto sucede en medicina. Para realizar un electrocardiograma se sitúan electrodos en la superficie del pecho cerca del corazón. Para realizar un electroencefalograma se colocan electrodos en la superficie de la cabeza.

Sin embargo, por muy cerca que esté un sensor de una fuente, la señal que recogerá será una mezcla de las señales emitidas por todas las fuentes. Es decir, en medidas realizadas con varios sensores, los datos son mezclas de las fuentes que se pretenden conocer.

Por lo tanto, las fuentes que se pretendían estudiar quedan mezcladas entre sí (no pudiéndose estudiar por separado) y, a veces, ocultas por otras fuentes no deseadas.

Es, pues, deseable encontrar un método que permita separar las fuentes que se encuentran mezcladas en las observaciones, con el objetivo de estudiarlas por separado y eliminar las no deseadas. Este problema se denomina separación ciega de fuentes (BSS: Blind Source Separation). Se denomina ciega porque no se dispone de datos a priori sobre cada una de las fuentes a separar.

En este proyecto se pretende estudiar la aplicación de la BSS en señales biomédicas. Para ello se supondrá que las fuentes que originan las señales son mutuamente independientes, que las mezclas son lineales e instantáneas y que el ruido aditivo que contienen es Gaussiano. Esto último se debe a que este tipo de ruido permite la separación de las fuentes.

Las aplicaciones biomédicas de la separación de fuentes son numerosas: electrocardiogramas (ECG), electroencefalogramas (EEG), magnetoencefalogramas (MEG), resonancias magnéticas funcionales (fMRI), etc.

Veremos que existe un método alternativo a la separación: la extracción ciega de fuentes (BSE: Blind Source Extraction). Consiste en extraer sólo un cierto número de fuentes, y no todas. Este método se puede combinar con la deflación (eliminar de los datos las componentes ya extraídas) para obtener todas las fuentes.

1.1 BSS Y BSE

En el problema de BSS las mezclas pueden ser lineales (las observaciones son, en este caso, combinaciones lineales de las fuentes) o no. En este proyecto se estudian mezclas lineales, ya que en la mayoría de las medidas de biomedicina la aproximación de mezclas lineales es buena.

Las mezclas pueden ser también instantáneas o convolutivas. En el primer caso las salidas en un determinado instante sólo dependen de las entradas en ese instante. Cuando las mezclas son convolutivas las salidas dependen también de las entradas en instantes anteriores. Es decir, el sistema tiene memoria. En muchos problemas de BSS (como en separación de hablantes) las mezclas han de suponerse convolutivas, pero en nuestro caso (y debido a la alta velocidad de propagación de las señales en biomedicina con respecto a sus máximas frecuencias) se pueden suponer mezclas instantáneas.

Una última distinción que se realiza en cuanto al tipo de mezclas en un problema de BSS es la consideración o no de ruido aditivo. En realidad se suele considerar un modelo sin ruido aditivo y suponer que el ruido aditivo que pueda haber será Gaussiano.

Cuando se considera un modelo sin ruido aditivo y de mezclas lineales e instantáneas el vector de observaciones se obtiene multiplicando una matriz de mezclas de elementos constantes por el vector de fuentes. La separación consiste en hallar esa matriz de mezclas y, a partir de ella, hallar la matriz de separación (su inversa), por la que hay que multiplicar el vector de observaciones para obtener las fuentes. Por lo tanto, para poder llevar a cabo la separación es necesario que la matriz de mezclas sea invertible.

Imponiendo la independencia mutua de las salidas (las señales obtenidas tras aplicar la matriz de separación) encontraremos las fuentes en cada una de las salidas. Sin embargo, en presencia de no más de una fuente Gaussiana, la independencia 2 a 2 de las salidas garantiza la separación. Esta independencia se verifica mediante la utilización de estadísticos de orden superior a 2 (HOS). Más concretamente, si todos los cumulantes cruzados (emparejando las salidas 2 a 2) posibles son nulos, se produce la independencia 2 a 2. Sin embargo, si hay más de una fuente no Gaussiana, pueden aparecer esas fuentes Gaussianas combinadas en las salidas de forma que sus correlaciones cruzadas (sus cumulantes cruzados de orden 2) sean nulas, por lo que habrá independencia 2 a 2 de las salidas (los cumulantes de orden superior a 2 de señales Gaussianas son nulos) sin que exista separación de las fuentes Gaussianas (de ahí que si hay más de una fuente Gaussiana la independencia 2 a 2 de las salidas sólo garantice la separación de las fuentes no Gaussianas).

Por otra parte (y teniendo en cuenta el uso de un modelo que ignora el ruido aditivo), si existe ruido aditivo Gaussiano (lo cual sería equivalente a la existencia del doble de fuentes de las consideradas), en cada salida (para que exista independencia de salidas 2 a 2) habrá una fuente (no Gaussiana) y una combinación lineal de las fuentes de ruido (Gaussianas). Por eso se considera un modelo sin ruido y un ruido Gaussiano.

Hay que decir que en mezclas lineales no es necesaria la utilización de todos los cumulantes para garantizar la independencia. Basta con un conjunto limitado.

También hay que tener en cuenta que no es suficiente la anulación de las correlaciones cruzadas (la diagonalización de la matriz de correlaciones cruzadas), ya que esta condición no proporciona un número suficiente de ecuaciones.

La diagonalización de la matriz de correlaciones cruzadas es un paso previo que se utiliza (se denomina decorrelación de las observaciones) para conseguir una matriz de separación ortogonal, lo cual es utilizado por los algoritmos de separación.

Una forma concreta de realizar la decorrelación de las observaciones es el método PCA (Análisis de Componentes Principales). Este método tiene la particularidad de que las salidas procedentes de la aplicación del método, además de estar decorreladas, quedan ordenadas en función de su importancia (del valor absoluto de sus autovalores). Esto permite reducir el número de observaciones manteniendo en gran parte la información que tenía el conjunto de todas las observaciones originales.

Existe un enfoque alternativo a la BSS, que es el Análisis de Componentes Independientes (ICA). Se trata de un método aditivo, frente a BSS, que es un método multiplicativo. Consiste en considerar el vector de las observaciones como una suma de vectores. Se busca que en el vector de las salidas las componentes (los vectores que se suman para dar lugar al vector de salidas) sean lo más independientes posible.

Para realizar algoritmos de separación es preciso establecer criterios de separación. La forma de establecer criterios de separación es buscar funciones contraste. Se trata de funciones cuyo argumento es el vector de salidas (considerado como una variable aleatoria multidimensional) y cuyo máximo se produce cuando los elementos del vector son independientes. De esta manera se obtienen criterios como el de mínima información mutua (MMI), el de máxima verosimilitud (ML) o el INFOMAX (o ENTMAX, de maximización de la información o de la entropía).

El criterio MMI trata de minimizar la desigualdad existente entre la f.d.p. (función de densidad de probabilidad) del vector de salidas y el producto de las f.d.p. de los elementos de ese vector. El criterio ML trata de maximizar la probabilidad de obtener las observaciones obtenidas condicionada a la matriz de separación elegida (así trata de encontrar la matriz de separación idónea). El criterio ENTMAX (ME) trata de maximizar la información mutua de las entradas y las salidas (la cantidad de información de las salidas que no proviene de las entradas). Estos tres criterios coinciden localmente, aunque globalmente sólo coinciden el ML y el ME.

La BSE parte de las mismas ideas que la BSS, pero, en lugar de buscar la separación de todas las fuentes mediante una matriz de separación cuadrada, busca solamente la obtención (extracción) de un conjunto de las fuentes. Puesto que en principio no se sabe qué fuentes serán las extraídas es necesario buscar algoritmos que extraigan fuentes de interés. Es decir, fuentes lo menos Gaussianas posible. Una fuente es poco Gaussiana si el valor absoluto de su kurtosis (cumulante de orden cuatro) es alto. Se denomina superGaussiana si la kurtosis es positiva y subGaussiana si es negativa.

La BSE se puede utilizar también junto con otro proceso denominado deflación (consistente en eliminar de los datos las fuentes extraídas) para extraer paso a paso todas las fuentes. Se realiza una extracción y posteriormente una deflación. Se repite este proceso hasta llegar al último grupo de fuentes.

1.2 SEÑALES BIOLÓGICAS

Las señales procedentes de mediciones con varios sensores en biomedicina cumplen, en general, las condiciones de mezclas lineales e instantáneas y la independencia de las fuentes. Los dos problemas que se pretenden resolver mediante la utilización de la BSS son la eliminación de artefactos (señales procedentes de fuentes no deseadas que aparecen mezcladas con las señales de interés) y el propio análisis de las señales de interés (la separación de las distintas fuentes para su estudio por separado).

Como se ha dicho, existen numerosas aplicaciones de BSS en medicina. Una de las más estudiadas es la extracción del ECG del feto de una mujer embarazada. En las medidas originales (las observaciones) aparecen mezclados el ECG de la madre y el del feto (que es más débil), además de otros artefactos, por lo que es necesario separarlos para una correcta visualización del ECG del feto.

Otra aplicación de BSS en el campo del ECG es la separación de las componentes independientes que se encuentran en señales consistente en la evolución temporal de parámetros del ECG (los intervalos RR y QT), para el estudio del sistema nervioso autónomo.

La aplicación de BSS en fMRI se puede enfocar desde dos puntos de vista. Una posibilidad es considerar cada imagen como un sensor con un cierto número de muestras (los puntos de que consta la imagen). El otro punto de vista consiste en considerar que cada punto de la imagen es un sensor y cada imagen es una muestra de cada sensor.

La aplicación más importante de BSS en medicina es su utilización en EEG y MEG. La BSS permite (como siempre) eliminar artefactos, pero, además tiene una gran utilidad para el estudio de respuestas cerebrales a estímulos simultáneos, con lo que permite un estudio mucho más amplio del comportamiento cerebral que el que se podría llevar a cabo sin el empleo de BSS, ya que, gracias a la separación de las fuentes que originan las respuestas, es posible el estudio por separado de cada una de ellas (sabiendo que han sido realizadas en presencia de varios estímulos).

Otra utilidad de la BSS es la localización de las fuentes. Una vez obtenido el vector de extracción de una componente (directamente mediante BSE o utilizando solamente la fila de la matriz de separación correspondiente a esa componente), su

pesudoinversa es la proyección de la componente en los sensores. Conociendo la localización de los sensores es posible representar mapas que permitan una localización de la fuente (aquella zona en que la proyección de la componente sea mayor).

La aplicación de BSS a EEG y MEG es importante porque EEG y MEG son técnicas no invasivas (permiten el estudio del comportamiento de la corteza cerebral desde el exterior de la cabeza y sin perturbar ese comportamiento).

Una aplicación, todavía en estudio, de BSS sobre los datos de EEG o MEG es la identificación y localización de tumores cerebrales.

1.3 ALGORITMOS Y SOFTWARE

En este proyecto se emplearán 4 algoritmos (algunos de ellos con distintas variantes) para resolver un problema de extracción de artefactos en un MEG. Los datos proceden de 122 sensores y tienen una longitud de 17730 muestras, correspondientes a 2 minutos de grabación. Esto produce una gran lentitud de procesado, por lo que una de las cualidades a comparar en los algoritmos es el tiempo de ejecución.

Tres de los algoritmos utilizados son de extracción, mientras que el cuarto es de separación. Dos de los algoritmos de extracción están basados en los cumulantes cruzados. El tercero es una particularización (con ligeras diferencias) de uno de los dos anteriores (está basado en la kurtosis de las salidas), aunque posee otras variantes. El algoritmo de separación está basado en el criterio INFOMAX y da lugar a dos variantes: algoritmo INFOMAX sencillo y algoritmo ICA extendido.

Para la realización de los análisis se emplearán distintos programas para MATLAB. Se utilizará también para la representación de los datos y de los resultados, así como para la localización de las fuentes, el toolbox para MATLAB ICA/EEG.

1.4 ESTRUCTURA DEL PROYECTO

Este proyecto consta de 7 capítulos. En el capítulo 2 se realiza una descripción de la BSS y sus condiciones, de los casos que se presentan y de los criterios existentes más importantes para resolver el problema. Se realiza una descripción del análisis de componentes independientes (ICA: Independent Components Analysis), que consiste en

un enfoque diferente a la hora de resolver el problema. Se describen también la extracción y el preblanqueo de las observaciones.

En el capítulo 3 se describen las distintas aplicaciones biomédicas de la separación ciega de fuentes. Las dos finalidades principales son la eliminación de fuentes indeseadas (artefactos) y la separación de las propias señales deseadas para posibilitar su estudio por separado.

En el capítulo 4 se realiza una descripción de los algoritmos que posteriormente serán utilizados en el capítulo de simulaciones. Se trata de tres algoritmos de extracción y uno de separación.

En el capítulo 5 se describen los programas de ordenador (archivos *.m para MATLAB) que se utilizarán en el capítulo de simulaciones.

En el capítulo 6 se detallan las simulaciones llevadas a cabo. Se trata de la separación y extracción de artefactos de los datos de un MEG de 122 canales y 2 minutos de duración.

El capítulo 7 se dedica a las conclusiones y líneas futuras.

CAPÍTULO 2

SEPARACIÓN Y EXTRACCIÓN CIEGA DE FUENTES

La mayor parte de este capítulo, así como su estructura están basadas en [Cruces99].

El problema de la separación ciega de fuentes (BSS) consiste en recuperar las fuentes que aparecen mezcladas en las observaciones de varias señales desconociendo las propiedades estadísticas de esas fuentes.

Éste es un problema muy difícil de resolver de forma general, por lo que hay que hacer aproximaciones para los diversos casos que aparecen. Una primera distinción es la realizada entre

- Mezcla **lineal**: en ella las observaciones son combinaciones lineales de las fuentes.
- Mezcla **no lineal**: las combinaciones de las fuentes para formar las observaciones son no lineales. Este caso es muy complicado y está poco estudiado matemáticamente. Los algoritmos que veremos consideran las mezclas lineales.

Otra clasificación es

- Mezcla **instantánea** el sistema de mezcla no tiene memoria. Luego las observaciones en un instante dado dependerán únicamente de las fuentes en ese instante. Si además el sistema es LTI, éste consistirá en una matriz invariante en el tiempo.

- Mezcla **convolutiva**: el sistema de mezcla tiene memoria. En este caso, además de la diversidad espacial, es necesario utilizar la diversidad temporal para resolver el problema.

Este capítulo se ocupa de las mezclas instantáneas, puesto que los algoritmos que serán estudiados posteriormente suponen este tipo de mezclas.

Una tercera clasificación es

- mezclas **sin ruido**
- mezclas **con ruido**: además de la combinación de las fuentes, en las observaciones aparece un ruido aditivo estadísticamente independiente de las fuentes.

2.1 MEZCLAS INSTANTÁNEAS

Este modelo es el más utilizado debido a su mayor sencillez matemática y su precisión en muchas aplicaciones.

2.1.1 Caso sin ruido

El modelo es el siguiente

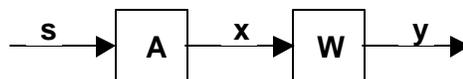


Figura 2-1 Modelo de mezcla y separación sin ruido

En este modelo $\mathbf{s} = (s_1[n], s_2[n], \dots, s_N[n])^T$ es el vector de fuentes, $\mathbf{x} = (x_1[n], x_2[n], \dots, x_N[n])^T$ el vector de observaciones e $\mathbf{y} = (y_1[n], y_2[n], \dots, y_N[n])^T$ el de salidas (en el que se deben encontrar las fuentes separadas).

La matriz \mathbf{A} ($N \times N$) es la matriz de mezclas. Por lo tanto $\mathbf{x} = \mathbf{A}\mathbf{s}$. La matriz \mathbf{W} ($N \times N$) es la de separación, por lo que $\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{A}\mathbf{s}$. Denominando \mathbf{G} (matriz global de transferencia, $N \times N$) al producto $\mathbf{W}\mathbf{A}$ se tiene que $\mathbf{y} = \mathbf{G}\mathbf{s}$.

Las fuentes $s_i[n]$, $i = 1, 2, \dots, N$ son realizaciones del proceso estocástico $S_i[n]$, $i = 1, 2, \dots, N$. Estas fuentes se suponen de media cero, mutuamente independientes y con

función de densidad de probabilidad (f.d.p.) desconocida $p_{s_i}(s_i)$. Debido a que son independientes se tiene que

$$p_s(\mathbf{s}) = \prod_{i=1}^N p_{s_i}(s_i)$$

2.1.2 Caso con ruido

En este caso el modelo es ligeramente distinto

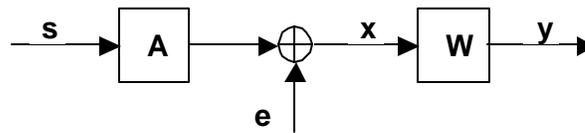


Figura 2-2 Modelo de mezclas y separación con ruido

El vector $\mathbf{e} = (e_1[n], e_2[n], \dots, e_N[n])^T$ es el vector de ruido aditivo. En este caso las observaciones son $\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{e}$ y las salidas son $\mathbf{y} = \mathbf{W}\mathbf{A}\mathbf{s} + \mathbf{W}\mathbf{e}$. Considerando de nuevo $\mathbf{G} = \mathbf{W}\mathbf{A}$ tenemos que $\mathbf{y} = \mathbf{G}\mathbf{s} + \mathbf{W}\mathbf{e}$. Por lo tanto, a las fuentes hay que añadirles N nuevas fuentes, que son las fuentes de ruido.

Podemos así formar un vector de fuentes ampliado $\mathbf{s}' = (\mathbf{s}^T | \mathbf{e}^T)^T$ de dimensión $2N \times 1$, y una matriz de mezclas ampliada $\mathbf{A}' = (\mathbf{A} | \mathbf{I})$ de dimensión $N \times 2N$, donde \mathbf{I} es la matriz identidad $N \times N$.

De esta manera el vector de observaciones se puede expresar como el producto de la matriz ampliada de mezclas por el vector de observaciones ampliado. Es decir, $\mathbf{x} = \mathbf{A}'\mathbf{s}'$, y es de dimensión $N \times 1$.

Esto significa que tenemos el doble de fuentes que de observaciones, por lo que habrá que añadir información para poder recuperar las fuentes. Al tener más fuentes que observaciones lo máximo que se puede conseguir es obtener en cada salida $y_i[n]$ una combinación lineal de un grupo distinto de fuentes. Los grupos serán disjuntos (ninguna fuente pertenecerá a más de un grupo) debido a la independencia de las fuentes. Esto se consigue imponiendo la independencia de las salidas.

Las aproximaciones que se llevan a cabo para tratar el caso con ruido consisten en considerar el ruido Gaussiano y emplear estadísticos de orden superior. Esto se debe

a que muchos estadísticos de orden superior son invariantes frente a la adición de ruido Gaussiano, debido a que esos estadísticos son nulos para señales Gaussianas. Por ello muchos algoritmos pensados para el caso sin ruido (como los algoritmos que utilizaremos en este proyecto) son válidos para llevar a cabo la separación incluso en presencia de ruido Gaussiano.

2.1.3 Separabilidad e identificabilidad

Separabilidad es la posibilidad de llevar a cabo la separación. Sólo depende de la matriz de mezclas \mathbf{A} . Si \mathbf{A} es invertible la separación es posible ($\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$). En caso contrario no es posible separar las fuentes.

Identificabilidad (también llamada principios de separación) es la capacidad de determinar la matriz de mezclas \mathbf{A} para posteriormente invertirla. La determinación de \mathbf{A} de manera exacta nunca será posible, por lo que siempre existirá un error. Este error aparecerá en la desviación de \mathbf{G} con respecto a la matriz identidad \mathbf{I} (si \mathbf{A} pudiese ser determinada de forma exacta \mathbf{W} sería \mathbf{A}^{-1} , por lo que \mathbf{G} sería la identidad).

Para obtener los principios de separación se utiliza el teorema de Darmois-Skitovich.

Teorema 2.1. (de Darmois-Skitovich): Sea $\mathbf{s} = (s_1[n], s_2[n], \dots, s_N[n])^T$ un vector aleatorio ($N \geq 2$) cuyas componentes son mutuamente independientes y sean 2 salidas independientes

$$\begin{aligned} y_1[n] &= G_{11}s_1[n] + G_{12}s_2[n] + \dots + G_{1N}s_N[n] \\ y_2[n] &= G_{21}s_1[n] + G_{22}s_2[n] + \dots + G_{2N}s_N[n] \end{aligned}$$

Si para un índice arbitrario i se cumple $G_{i1} \neq 0$ y $G_{i2} \neq 0$ entonces $s_i[n]$ tiene una distribución Gaussiana.

También se utiliza el siguiente lema

Lema 2.1.: Si en una matriz no singular $\mathbf{G} \in \mathbb{R}^{N \times N}$ la primera fila es ortogonal al resto entonces $G_{1k} = 0 \quad k = 2, 3, \dots, N$ si y sólo si $G_{ki} = 0 \quad k = 2, 3, \dots, N$.

Por otra parte \mathbf{G} es ortogonal. Esto se consigue por la decorrelación de las salidas: $\mathbf{R}_{yy} = \mathbf{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{I}$. $\mathbf{R}_{yy} = \mathbf{G}\mathbf{R}_{ss}\mathbf{G}^T = \mathbf{G}\mathbf{I}\mathbf{G}^T$. $\mathbf{R}_{ss} = \mathbf{I}$ porque al ser las fuentes independientes también son incorreladas.

Empleando el teorema de Darmois-Skitovich y el lema anterior se deduce el siguiente corolario

Corolario 2.1.: *Si todas las fuentes menos una son no Gaussianas entonces la independencia 2 a 2 de las salidas garantiza la separación.*

Supongamos que $s[n]$ es la fuente Gaussiana. Si las salidas son independientes 2 a 2, como las fuentes son no Gaussianas excepto $s[n]$, en cada columna de la submatriz $[G_{ij}]$, $i = 1, \dots, N$, $j = 2, \dots, N$ sólo hay un elemento distinto de 0. Como hay más filas que columnas tenemos al menos una fila de ceros. Podemos colocar una de esas filas en la primera posición (lo único que cambiará será la posición de las salidas). Aplicando el lema veremos que la primera columna tiene todos sus elementos (excepto el primero) nulos. Sin embargo, si hubiese más de una fila de ceros en la submatriz antes definida, debido a la columna de ceros existente que acabamos de comprobar habrá una fila de ceros en la matriz \mathbf{G} . Esto entra en contradicción con el hecho de que \mathbf{G} es ortogonal. Luego, en la submatriz definida hay exactamente 1 fila de ceros, por lo que los elementos no nulos están cada uno en una fila. Al aplicar el lema vemos que en cada fila y en cada columna de \mathbf{G} hay exactamente 1 elemento no nulo. Reordenando las filas (las salidas) y recordando que \mathbf{G} es ortogonal tenemos que $\mathbf{G} = \mathbf{I}$. Es decir, se produce la separación.

Por lo tanto, una forma de diseñar algoritmos de separación es forzar la independencia estadística 2 a 2 de las salidas. Ésta es una condición más débil que la independencia mutua.

Hay que notar que la separación se produce por la independencia de las salidas, no por su decorrelación (la cual sólo nos asegura la ortogonalidad de \mathbf{G}).

Veamos el siguiente corolario

Corolario 2.2.: *En presencia de más de una fuente Gaussiana la propiedad de independencia 2 a 2 de las salidas sólo garantiza la separación de las fuentes no Gaussianas.*

Supongamos que tenemos L fuentes Gaussianas. El corolario nos muestra que si las fuentes y las salidas tienen varianza unidad, podemos permutar \mathbf{G} de manera que se

descomponga como $\mathbf{G} = \text{diag}\{\mathbf{I}, \mathbf{U}\}$, donde \mathbf{I} es la matriz identidad $(N-L) \times (N-L)$ y \mathbf{U} es una matriz unitaria $L \times L$.

Matriz global de transferencia transparente (\mathbf{G}): es aquella que logra la separación de las fuentes.

Esta matriz no es única. Obviamente, para recuperar las fuentes en el mismo orden en que se agrupan en el vector de fuentes y con su misma varianza es necesario que $\mathbf{G} = \mathbf{I}$. Pero, una reorganización de las salidas (permutación de \mathbf{G}) o un escalado de las salidas (escalado de las filas de \mathbf{G}) no alteran la independencia de las mismas. Por lo tanto el conjunto de soluciones al problema de BSS vendrá dado por $\mathbf{G} = \mathbf{L}\mathbf{P}$, siendo $\mathbf{L} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ y \mathbf{P} cualquier matriz de permutación.

Esto quiere decir que podemos recuperar las fuentes sin orden (lo cual no es un grave problema, ya que el orden en el vector de fuentes es arbitrario) y escaladas. Para identificar (reconocer) cada fuente habrá que observar algunas propiedades de las salidas, como el histograma, la forma de onda, etc.

2.1.4 Independencia y HOS

Sea $\mathbf{V} = (Y_i, Y_j)^T$. Decir que Y_i e Y_j son independientes es equivalente a decir que

$$p_{\mathbf{V}}(\mathbf{v}) = p_{Y_i}(y_i)p_{Y_j}(y_j)$$

En la práctica este criterio no es útil por la dificultad de estimar las f.d.p. de las salidas. Debido a ello se utilizan las funciones características.

La primera función característica de una variable aleatoria es el conjugado de la transformada de Fourier del conjugado de su f.d.p.:

$$\phi_{\mathbf{V}}(\mathbf{w}) = \mathbb{E}\left[e^{j\mathbf{w}^T\mathbf{v}}\right] = \int_{-\infty}^{\infty} p_{\mathbf{V}}(\mathbf{v})e^{j\mathbf{w}^T\mathbf{v}} d\mathbf{v}$$

La segunda función característica es

$$\psi_{\mathbf{V}}(\mathbf{w}) = \ln[\phi_{\mathbf{V}}(\mathbf{w})]$$

Es equivalente el hecho de que Y_i e Y_j sean independientes al hecho de que

$$\phi_{\mathbf{V}}(\mathbf{w}) = \phi_{Y_i}(\omega_i)\phi_{Y_j}(\omega_j)$$

que a su vez es equivalente a decir que

$$\psi_{\mathbf{V}}(\mathbf{w}) = \psi_{Y_i}(\omega_i) + \psi_{Y_j}(\omega_j)$$

Veamos: por una parte si Y_i e Y_j son independientes entonces

$$p_{\mathbf{V}}(\mathbf{v}) = p_{Y_i}(y_i)p_{Y_j}(y_j)$$

Esto conduce a que

$$\begin{aligned}\phi_{\mathbf{V}}(\mathbf{w}) &= \int_{-\infty}^{\infty} p_{\mathbf{V}}(\mathbf{v}) e^{j\omega_i y_i} e^{j\omega_j y_j} dy_i dy_j = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{Y_i}(y_i) p_{Y_j}(y_j) e^{j\omega_i y_i} e^{j\omega_j y_j} dy_i dy_j = \\ &= \phi_{Y_i}(\omega_i) \phi_{Y_j}(\omega_j)\end{aligned}$$

Por otra parte, si

$$\phi_{\mathbf{V}}(\mathbf{w}) = \phi_{Y_i}(\omega_i) \phi_{Y_j}(\omega_j)$$

entonces

$$\int_{-\infty}^{\infty} [p_{\mathbf{V}}(\mathbf{v}) - p_{Y_i}(y_i)p_{Y_j}(y_j)] e^{j\omega_i y_i} e^{j\omega_j y_j} dy_i dy_j = 0$$

Es decir

$$F [p_{\mathbf{V}}(\mathbf{v}) - p_{Y_i}(y_i)p_{Y_j}(y_j)](\omega_i, \omega_j) = 0$$

Por lo tanto

$$p_{\mathbf{V}}(\mathbf{v}) = p_{Y_i}(y_i)p_{Y_j}(y_j)$$

Con lo que Y_i e Y_j son independientes.

Por último

$$\phi_{\mathbf{V}}(\mathbf{w}) = \phi_{Y_i}(\omega_i) \phi_{Y_j}(\omega_j) \Rightarrow \psi_{\mathbf{V}}(\mathbf{w}) = \ln[\phi_{Y_i}(\omega_i) \phi_{Y_j}(\omega_j)] = \psi_{Y_i}(\omega_i) + \psi_{Y_j}(\omega_j)$$

Y

$$\psi_{\mathbf{V}}(\mathbf{w}) = \psi_{Y_i}(\omega_i) + \psi_{Y_j}(\omega_j) \Rightarrow \phi_{\mathbf{V}}(\mathbf{w}) = e^{\psi_{Y_i}(\omega_i)} e^{\psi_{Y_j}(\omega_j)} = \phi_{Y_i}(\omega_i) \phi_{Y_j}(\omega_j)$$

Expandiendo en serie de Taylor la ecuación

$$\psi_{\mathbf{V}}(\mathbf{w}) = \psi_{Y_i}(\omega_i) + \psi_{Y_j}(\omega_j)$$

vemos que los términos cruzados $\omega_i^\alpha \omega_j^\beta$ se deben anular, ya que no aparecen en el término de la derecha. Los coeficientes de estos términos son proporcionales a los estadísticos denominados **cumulantes**.

$$C_{y_i, y_j}^{\alpha, \beta} = (-j)^{\alpha + \beta} \frac{\partial^{\alpha + \beta} \Psi_{\mathbf{V}}(\omega_i, \omega_j)}{\partial \omega_i^\alpha \partial \omega_j^\beta} \Bigg|_{\omega_i = \omega_j = 0} \quad \text{cumulante cruzado de orden } \alpha + \beta$$

Por lo tanto, la independencia de Y_i e Y_j significa

$$C_{y_i, y_j}^{\alpha, \beta} = 0 \quad \forall \alpha \neq \beta, \alpha, \beta \in \mathbb{N}^+$$

Definimos la matriz de cumulantes cruzados como

$$\mathbf{C}_{yy}^{\alpha\beta} = \left[C_{y_i, y_j}^{\alpha, \beta} \right]_{i, j}$$

Con esta definición se tiene que

$$Y_i, Y_j \text{ indeps.} \Leftrightarrow \mathbf{C}_{yy}^{\alpha\beta} \text{ diagonal } \forall \alpha, \beta \in \mathbb{N}^+$$

Pero, en el caso de las mezclas lineales un conjunto limitado de cumulantes cruzados proporcionan suficientes ecuaciones para determinar la matriz de mezclas.

No son, sin embargo, suficientes los cumulantes de orden 2 (la decorrelación de las salidas no implica su independencia). Es decir, no es suficiente la ecuación

$$\mathbf{C}_{yy}^{11} = \mathbf{I}$$

Esta ecuación sólo proporciona $N(N-1)$ ecuaciones, mientras que la matriz de separación consta de N^2 elementos. Por lo tanto son necesarios cumulantes cruzados de orden mayor que 2 (estadísticos de orden superior, HOS).

No obstante, la decorrelación de las salidas se utiliza para simplificar el problema. Muchos algoritmos realizan, como paso previo a la separación, el preblanqueo (prewhitening o sphering) o decorrelación de las observaciones. Esto hace que la matriz de separación sea unitaria, hecho aprovechado posteriormente por estos algoritmos. El preblanqueo se realiza de la siguiente manera.

Sea \mathbf{x} el vector de las observaciones. Si normalizamos la varianza de cada componente, la matriz de correlaciones será $\mathbf{R}_{xx} = E[\mathbf{x}\mathbf{x}^H]$. Esta matriz se puede descomponer como $\mathbf{R}_{xx} = \mathbf{U}\mathbf{L}\mathbf{U}^H$, donde \mathbf{U} es la matriz cuyas columnas son los autovectores y \mathbf{L} la matriz diagonal (si todos son distintos, lo cual ocurre normalmente en la realidad) de los autovalores ($\mathbf{L} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$). La matriz de preblanqueo (que es ortogonal) se define como $\mathbf{B} = \mathbf{L}^{-1/2}\mathbf{U}^H$ donde $\mathbf{L}^{-1/2} = \text{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \dots, \lambda_N^{-1/2})$. Por lo tanto las observaciones preblanqueadas serán $\mathbf{x}' = \mathbf{B}\mathbf{x}$. Veamos que esto es así: la

nueva matriz de correlaciones será $\mathbf{R}_{\mathbf{X}\mathbf{X}'} = \mathbf{B}\mathbf{R}_{\mathbf{X}\mathbf{X}'}\mathbf{B}^H = \mathbf{L}^{-1/2}\mathbf{U}^H\mathbf{U}\mathbf{L}\mathbf{U}^H\mathbf{U}\mathbf{L}^{-1/2} = \mathbf{L}^{-1/2}\mathbf{L}\mathbf{L}^{-1/2} = \mathbf{I}$ ($\mathbf{U}^H\mathbf{U} = \mathbf{I}$ porque los autovectores son ortogonales).

Los cumulantes cruzados se pueden expresar en función de los momentos cruzados, más sencillos de calcular.

$$\begin{aligned} M_{y_i, y_j}^{p, q} &= E[y_i^p y_j^q] \\ C_{y_i, y_j}^{1, 1} &= M_{y_i, y_j}^{1, 1} \\ C_{y_i, y_j}^{3, 1} &= M_{y_i, y_j}^{3, 1} - 3M_{y_i, y_j}^{2, 0} M_{y_i, y_j}^{1, 1} \\ C_{y_i, y_j}^{2, 2} &= M_{y_i, y_j}^{2, 2} - M_{y_i, y_j}^{2, 0} M_{y_i, y_j}^{0, 2} - 2(M_{y_i, y_j}^{1, 1})^2 \\ C_{y_i, y_j}^{1, 3} &= M_{y_i, y_j}^{1, 3} - 3M_{y_i, y_j}^{0, 2} M_{y_i, y_j}^{1, 1} \end{aligned}$$

De nuevo hemos visto que si más de una fuente es Gaussiana la independencia de las salidas no garantiza la separación de las fuentes Gaussianas. En efecto, siempre que en las salidas aparezcan las fuentes Gaussianas combinadas de tal manera que sus correlaciones cruzadas sean nulas se garantiza la independencia (puesto que las señales Gaussianas tienen los cumulantes de orden mayor que 2 nulos), y sin embargo las fuentes Gaussianas aparecen combinadas en las salidas. Es decir, existe independencia de las salidas sin que exista separación de las fuentes Gaussianas.

2.1.5 ICA

El procedimiento de preblanqueo que hemos visto en el apartado anterior es el conocido como **PCA** (Análisis de componentes principales) [Bell2000]. Este procedimiento busca la ortogonalización de las observaciones (su decorrelación), lo cual, como hemos visto, no garantiza la independencia ni, por lo tanto, la separación de las fuentes en esas observaciones preblanqueadas (sin embargo, muchas veces el proceso de preblanqueo ya permite vislumbrar la forma de algunas de las fuentes). Un uso frecuente de PCA (aparte del citado en el apartado anterior) es la reducción de la dimensión del vector de las observaciones. Ello se consigue tomando $\mathbf{L}' = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ y $\mathbf{U} = [\mathbf{u}_1 | \mathbf{u}_2 | \dots | \mathbf{u}_n]$ con $i \leq N$. Para conservar las componentes más importantes los autovalores se escogen en orden decreciente de su valor absoluto.

El **ICA** (Análisis de componentes independientes) surgió como una generalización de PCA y supone un punto de vista más amplio del que hemos visto

hasta ahora de resolver el problema de BSS. El ICA persigue maximizar la independencia entre las componentes que forma las observaciones. Al contrario que BSS no precisa de la existencia de una matriz de mezclas. Además, mientras que BSS es un modelo de separación multiplicativo ($\mathbf{x} = \mathbf{A}\mathbf{s}$, $\mathbf{y} = \mathbf{W}\mathbf{x}$), el ICA es un modelo aditivo. Si descomponemos \mathbf{A} en sus columnas $\mathbf{A} = [\mathbf{a}_1|\mathbf{a}_2|\dots|\mathbf{a}_N]$, las componentes de \mathbf{x} serán $\mathbf{x}_i = \mathbf{a}_i s_i$, de modo que

$$\mathbf{x} = \sum_{i=1}^N \mathbf{x}_i$$

El objetivo es descomponer \mathbf{x} en sus componentes más independientes \mathbf{x}_i $i = 1, 2, \dots, N$. De este modo desaparecen las indeterminaciones del problema, ya que los parámetros buscados no son los elementos de la matriz de mezcla (que no tiene por qué existir), sino las proyecciones \mathbf{P}_i que obtienen cada componente \mathbf{x}_i a partir de \mathbf{x}

$$\mathbf{P}_i = \frac{\mathbf{a}_i \mathbf{a}_i^T}{\mathbf{a}_i^T \mathbf{a}_i}$$

También existe MICA (ICA multidimensional), que es un caso más general consistente en buscar conjuntos de componentes multidimensionales independientes.

2.2 CRITERIOS DE SEPARACIÓN

Existen numerosos criterios de separación, aunque todos ellos están relacionados entre sí.

Veremos ahora cómo verificar la validez de esos criterios.

2.2.1 Funciones contraste

Fue Comon quien definió las funciones contraste.

Sea la variable aleatoria multidimensional \mathbf{Y} . Una función contraste es una correspondencia Ψ que asocia un número real $\Psi(\mathbf{Y})$ a esta variable aleatoria y que cumple las siguientes condiciones:

1. $\Psi(\mathbf{Y})$ es invariante a escalados invertibles

$$\forall \mathbf{L} \text{ matriz diagonal regular } \Psi(\mathbf{L}\mathbf{Y}) = \Psi(\mathbf{Y})$$

2. $\Psi(\mathbf{Y})$ sólo depende de la f.d.p. de \mathbf{Y} y es simétrica respecto de sus componentes Y_i

$$\forall \mathbf{P} \text{ matriz de permutación } \Psi(\mathbf{PY}) = \Psi(\mathbf{Y})$$

3. $\Psi(\mathbf{Y})$ sólo alcanza su máximo cuando las componentes de \mathbf{Y} son independientes.

$$\forall \mathbf{S} \text{ variable aleatoria de componentes independientes y } \forall \mathbf{G} \text{ regular}$$

$$\Psi(\mathbf{S}) \geq \Psi(\mathbf{Y}) \quad \text{donde } \mathbf{Y} = \mathbf{GS}$$

Como vimos en los corolarios al Teorema de Darrois-Skitovich (y suponiendo que sólo existe 1 fuente Gaussiana) la independencia de las salidas suponía la separación de las fuentes. Además no influye el escalado de las mismas o su permutación. Por lo tanto, cuando pueda encontrarse una función contraste, resolver el problema de BSS significa buscar el máximo global de esa función contraste.

La mayoría de las funciones contraste basadas en cumulantes utilizan la hipótesis de preblanqueo de las observaciones y decorrelación de las salidas, lo cual conduce a una matriz de separación, \mathbf{W} , ortogonal.

$$\mathbf{C}_{xx}^{1,1} = \mathbf{I}$$

$$\mathbf{C}_{yy}^{1,1} = \mathbf{I}$$

$$\mathbf{I} = \mathbf{C}_{yy}^{1,1} = \mathbf{WC}_{xx}^{1,1}\mathbf{W}^H = \mathbf{WIW}^H = \mathbf{WW}^H$$

Veamos el siguiente teorema

Teorema 2.2.: *La suma de los cuadrados de todos los cumulantes de orden r para una variable aleatoria \mathbf{Y} de media cero*

$$S_r = \sum_{i_1, i_2, \dots, i_r} (\mathbf{C}_{y_{i_1} y_{i_2} \dots y_{i_r}}^{1,1, \dots, 1})^2$$

es invariante bajo transformaciones ortogonales.

Este teorema es aplicable a las salidas en el caso de preblanqueo de las observaciones y decorrelación de las salidas, ya que en ese caso \mathbf{W} es ortogonal.

En ese caso la maximización de los elementos diagonales del tensor ($i_1 = i_2 = \dots = i_r$) equivale a la minimización de los cumulantes cruzados del mismo orden.

Esto nos lleva a que la función

$$\Psi(\mathbf{Y}) = \sum_{i=1}^N (C_{y_i}^r)^2 \quad \text{s. a. } \mathbf{W} = \mathbf{W}^T$$

es una función contraste si no existe más de 1 elemento diagonal nulo en el tensor cumulante de orden r de las fuentes \mathbf{S} .

Por comodidad consideremos las fuentes escaladas de manera que

$$\mathbf{C}_{ss}^{1,1} = \mathbf{I}$$

Entonces, tanto la matriz de mezclas \mathbf{A} como la matriz global \mathbf{G} serán ortogonales. \mathbf{G} cumplirá así la siguiente propiedad

$$\sum_{j=1}^N |G_{ij}|^2 = \sum_{i=1}^N |G_{ij}|^2 = 1$$

Veamos que $\Psi(\mathbf{Y})$ (donde $\mathbf{Y}=\mathbf{GS}$) es una función contraste:

1. Si $\mathbf{G} = \mathbf{L}$ diagonal, como \mathbf{G} es ortogonal $\mathbf{G} = \mathbf{I}$, por lo que no existe escalado y $\Psi(\mathbf{GS}) = \Psi(\mathbf{S})$ en caso de \mathbf{G} diagonal.
2. Los cumulantes no varían al variar la ordenación de sus argumentos, por lo que $\Psi(\mathbf{PS}) = \Psi(\mathbf{S})$, donde \mathbf{P} es una matriz de permutación.
3. Debido a la independencia de las fuentes y a las propiedades de los

$$\Psi(\mathbf{GS}) = \sum_{i=1}^N \left(\sum_{j=1}^N (G_{ij})^r C_{sj}^r \right)^2 \leq \sum_{j=1}^N (C_{sj}^r)^2 \left(\sum_{i=1}^N |G_{ij}|^r \right)^2$$

cumulantes

$$\sum_{i=1}^N |G_{ij}|^r \leq \sum_{i=1}^N |G_{ij}|^2 = 1$$

Para $r \geq 2$

Luego

$$\Psi(\mathbf{GS}) \leq \sum_{j=1}^N (C_{sj}^r)^2 \left(\sum_{i=1}^N |G_{ij}|^r \right)^2 \leq \sum_{j=1}^N (C_{sj}^r)^2 = \Psi(\mathbf{S})$$

En cada paso, el signo \leq es = sólo cuando \mathbf{G} es una matriz de permutación o bien la identidad. Es decir, en la separación.

Por lo tanto, $\Psi(\mathbf{Y})$ es una función contraste.

Una demostración similar nos mostraría que la función

$$\Psi(\mathbf{Y}) = \sum_{i=1}^N |C_{y_i}^r| \quad \text{s.a. } \mathbf{W} = \mathbf{W}^T$$

es una función contraste. Esta función, en el caso de que las fuentes sean de potencia unidad y kurtosis ($K_s = C_s^4$) negativas, se convierte en

$$\Psi(\mathbf{Y}) = -\sum_{i=1}^N E[y_i^4] \quad \text{s.a. } \begin{matrix} \mathbf{W} = \mathbf{W}^T \\ C_{y_i}^4 \leq 0 \quad \forall i=1, \dots, N \end{matrix}$$

2.2.2 Mínima información mutua

La información mutua (**MI**) es una forma de medir la dependencia de varias variables aleatorias. Se denota por $I(Y_1, Y_2, \dots, Y_N)$ y veremos que su opuesto es una función contraste.

$$I(Y_1, Y_2, \dots, Y_N) = D_{P_{\mathbf{Y}} \left| \prod_{i=1}^N P_{Y_i} \right.}$$

Es decir, es la divergencia de Kullback-Leibler (K-L) entre la f.d.p. conjunta y el producto de las f.d.p. marginales.

La divergencia K-L entre una f.d.p. $f_{\mathbf{Y}}(\mathbf{y})$ y otra $g_{\mathbf{Y}}(\mathbf{y})$ se define como

$$D_{f_{\mathbf{Y}}(\mathbf{y}) \left| g_{\mathbf{Y}}(\mathbf{y}) \right.} = \int_{-\infty}^{\infty} f_{\mathbf{Y}}(\mathbf{y}) \log \left(\frac{f_{\mathbf{Y}}(\mathbf{y})}{g_{\mathbf{Y}}(\mathbf{y})} \right) d\mathbf{y}$$

Esta función posee algunas propiedades de una métrica y permite así comparar funciones de densidad de probabilidad. Es una función cuasi-distancia entre 2 f.d.p.

Puesto que la independencia de variables se define como el hecho de que el producto de sus f.d.p. marginales sea igual a la f.d.p. conjunta, la información mutua mide cuán dependientes son las variables.

La divergencia K-L es invariante frente a escalados, permutaciones o transformaciones lineales monótonas aplicadas de forma pareja en sus argumentos. Esto hace que la MI sea invariante frente a escalados y permutaciones de las variables.

Además, por ser la divergencia K-L una función cuasi-distancia entre 2 f.d.p., es positiva excepto cuando esas 2 f.d.p. coinciden, en cuyo caso se anula. Esto nos lleva a

que $I(Y_1, \dots, Y_N) \geq 0$, cumpliéndose la igualdad si y sólo si Y_1, \dots, Y_N son independientes.

Por lo tanto, el opuesto de la información mutua es una función contraste que hay maximizar (o bien minimizar la propia información mutua) para conseguir salidas lo más independientes posible.

2.2.3 Máxima verosimilitud

El método de máxima verosimilitud (ML) se emplea para estimar un conjunto de parámetros de la f.d.p. de una variable aleatoria multidimensional. Se establece un modelo de f.d.p. para las observaciones $p_{\mathbf{x}}(\mathbf{X})$ que depende de un conjunto de parámetros \mathbf{x} . El método consiste en encontrar el conjunto de parámetros \mathbf{x} que con mayor probabilidad ha dado lugar a las observaciones. Es decir, encontrar \mathbf{x} que maximice la probabilidad $p_{\mathbf{x}}(\mathbf{X}|\mathbf{x})$. Así se define la función de verosimilitud

$$L(\mathbf{x}|\mathbf{X}) = p_{\mathbf{x}}(\mathbf{X}|\mathbf{x})$$

Esta función ha de ser maximizada. Como el máximo de una función no varía de posición (se produce para el mismo valor de sus argumentos) cuando se le aplica el logaritmo (por ser éste una función estrictamente monótona creciente), el problema se resuelve con el mismo resultado maximizando la función de verosimilitud logarítmica normalizada

$$l(\mathbf{x}|\mathbf{x}) = \frac{1}{T} \log [p_{\mathbf{x}}(\mathbf{x}|\mathbf{x})]$$

T es una constante arbitraria.

En el caso que nos ocupa (la determinación de la matriz de separación, \mathbf{W} , a partir de las observaciones) los parámetros son los coeficientes de la matriz \mathbf{W} . Los parámetros óptimos serán $\mathbf{W}_* = \mathbf{A}^{-1}$. Las observaciones cumplen $\mathbf{x} = \mathbf{W}_*^{-1}\mathbf{s}$ (el modelo es $\mathbf{x} = \mathbf{W}^{-1}\mathbf{s}$). La f.d.p. estimada de las fuentes es

$$p_{\mathbf{S}}(\mathbf{s})$$

Supongamos que tenemos un conjunto de T observaciones independientes (cada una de ellas multidimensional. Es decir, son realizaciones del experimento relacionado con la variable \mathbf{X}) $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$. La función de verosimilitud será

$$L(\mathbf{W}|\mathbf{x}) = p_{\hat{\mathbf{x}}}(\mathbf{x}|\mathbf{W})$$

Como las observaciones de la variable son independientes se tiene que

$$p_{\hat{\mathbf{x}}}(\mathbf{x}|\mathbf{W}) = \prod_{t=1}^T p_{\hat{\mathbf{x}}_t}(\mathbf{x}_t|\mathbf{W}) = \prod_{t=1}^T (p_{\hat{\mathbf{S}}}(\mathbf{W}\mathbf{x}_t)|\det(\mathbf{W})) = \prod_{t=1}^T (p_{\hat{\mathbf{S}}}(\mathbf{y}_t)|\det(\mathbf{W}))$$

Así pues

$$l(\mathbf{W}|\mathbf{x}) = \frac{1}{T} \log \left(\prod_{t=1}^T p_{\hat{\mathbf{S}}}(\mathbf{y}_t)|\det(\mathbf{W}) \right) = \frac{1}{T} \sum_{t=1}^T \log(p_{\hat{\mathbf{S}}}(\mathbf{y}_t)) + \log|\det(\mathbf{W})|$$

Puesto que el primero de los dos sumandos es la media muestral del término contenido en el sumatorio, y que esa media converge en probabilidad a su esperanza estadística (Ley de los grandes números) tenemos que

$$l(\mathbf{W}|\mathbf{x}) = E[\log(p_{\hat{\mathbf{S}}}(\mathbf{y}))] + \log|\det(\mathbf{W})|$$

Además

$$p_{\hat{\mathbf{S}}}(\mathbf{y}) = \frac{p_{\hat{\mathbf{S}}}(\mathbf{y})}{p_{\mathbf{Y}}(\mathbf{y})} p_{\mathbf{Y}}(\mathbf{y})$$

Y, debido a la independencia de las fuentes

$$p_{\hat{\mathbf{S}}}(\mathbf{y}) = \prod_{i=1}^N p_{\hat{\mathbf{S}}_i}(\mathbf{y}_i)$$

De este modo

$$l(\mathbf{W}|\mathbf{x}) = -D_{p_{\mathbf{Y}} \left| \prod_{i=1}^N p_{\hat{\mathbf{S}}_i} \right.} - h(\mathbf{X})$$

donde $h(\mathbf{X})$ es la entropía diferencial de las observaciones.

Como $h(\mathbf{X})$ es constante, la maximización de $l(\mathbf{W}|\mathbf{X})$ coincide con la minimización de la divergencia K-L entre la f.d.p. conjunta de las salidas y la f.d.p.

$$p_{\hat{\mathbf{S}}} = p_{\mathbf{S}}$$

conjunta estimada de las fuentes. Si el modelo es correcto entonces

con lo que la maximización de la función de verosimilitud logarítmica coincide localmente a la minimización de la información mutua.

2.2.4 Principios INFOMAX y ENTMAX

El problema de la separación ciega de fuentes o del análisis de componentes independientes se puede enfocar desde el punto de vista de la Teoría de la Información, disciplina que estudia los fenómenos relacionados con la comunicación.

En 1988 Linsker propuso como paradigma de aprendizaje no supervisado en una red neuronal la maximización de la transferencia de información entre las entradas y las salidas (principio INFOMAX). Bell y Sejnowski estudiaron en 1995 [Bell95] la aplicación del principio INFOMAX a la separación ciega de fuentes y en 1996 presentaron el principio ENTMAX como aplicación del principio INFOMAX a la BSS.

Hemos visto que el criterio de separación MMI buscaba minimizar la información mutua de las salidas, para hacerlas independientes. El criterio INFOMAX utiliza la función información mutua en otro sentido. Busca la maximización de la información de la información mutua de las entradas y las salidas. Es decir, la maximización de $I(\mathbf{Y};\mathbf{X})$. Definiendo la entropía diferencial de \mathbf{X} como

$$h(\mathbf{X}) = -E[\log(p_{\mathbf{X}}(\mathbf{x}))] = -\int_{\mathbf{x}} p_{\mathbf{X}}(\mathbf{x}) \log(p_{\mathbf{X}}(\mathbf{x})) d\mathbf{x}$$

y la entropía relativa condicional como

$$h(\mathbf{Y}|\mathbf{X}) = -E_{\mathbf{X}}[h(\mathbf{Y}|\mathbf{X} = \mathbf{x})] = -\int_{\mathbf{x}} \int_{\mathbf{y}} p_{\mathbf{X},\mathbf{Y}}(\mathbf{x};\mathbf{y}) \log(p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})) d\mathbf{y} d\mathbf{x}$$

se puede comprobar que

$$I(\mathbf{Y};\mathbf{X}) = h(\mathbf{Y}) - h(\mathbf{Y}|\mathbf{X})$$

La entropía es una medida de la incertidumbre o información asociada a una variable aleatoria (continua en el caso de la entropía diferencial).

La entropía relativa condicional es la cantidad de información de las salidas que no procede de las entradas.

Por lo tanto, la información mutua de las salidas y las entradas (la transferencia de información de las entradas a las salidas) es la diferencia entre la cantidad total de información asociada a las salidas y la cantidad de esta información que no procede de las entradas.

Cuando la relación entradas-salidas es determinista e invertible (como es \mathbf{W} en nuestro caso) $h(\mathbf{Y}|\mathbf{X})$ es constante e independiente de \mathbf{W} . La red neuronal de nuestro problema de BSS es de la forma $\mathbf{y} = \mathbf{W}\mathbf{x}$. Esto nos lleva a que

$$\frac{\partial I(\mathbf{Y}; \mathbf{X})}{\partial \mathbf{W}} = \frac{\partial h(\mathbf{Y})}{\partial \mathbf{W}}$$

Es decir, en BSS el principio ENTMAX (maximización de la entropía de las salidas o ME) es equivalente al principio INFOMAX. Sin embargo, $h(\mathbf{Y})$ (cumpliendo la relación $\mathbf{y} = \mathbf{W}\mathbf{x}$) diverge a infinito proporcionalmente con $\log|\det(\mathbf{W})|$, por lo que se maximiza no $h(\mathbf{Y})$ sino $h(\mathbf{Z})$, cumpliendo \mathbf{Z} la relación $\mathbf{z} = \mathbf{q}(\mathbf{y}) = [q_1(y_1), \dots, q_N(y_N)]^T$, donde

$$q_i(y_i) = \int_{-\infty}^{y_i} p_{S_i}(s_i) ds_i = P_{S_i}(y_i) \quad i = 1, \dots, N$$

quedando $q_i(y_i)$ acotada en el intervalo $[q_{\min}, q_{\max}]$.

Es decir, $q_i(y_i)$ es la función de densidad de probabilidad de S_i en y_i . Cuando las salidas coinciden con las fuentes, se tendrá una \mathbf{Z} uniforme.

Para cualquier variable aleatoria multidimensional (en particular para \mathbf{Z}) se cumple que

$$h(\mathbf{Z}) \leq \sum_{i=1}^N h(Z_i)$$

Además, para una variable aleatoria unidimensional se cumple que $h(Z_i) \leq \log(Z_i)$, siendo $Z_i = z_{i\max} - z_{i\min}$. Esto significa que el máximo de la entropía se produce para una variable uniforme.

Si $q_i(y_i)$, $i = 1, \dots, N$ están acotadas en el intervalo $[q_{\min}, q_{\max}]$ y, definiendo $\Delta = q_{\max} - q_{\min}$ se tiene que

$$h(\mathbf{Z}) \leq \sum_{i=1}^N h(Z_i) \leq \sum_{i=1}^N \log(\Delta) = N \cdot \log(\Delta)$$

Por lo tanto $h(\mathbf{Z})$ está acotada superiormente.

Es fácil ver que $h(\mathbf{Z})$ es el opuesto de la divergencia KL entre la f.d.p. de \mathbf{Z} y la f.d.p. de una variable aleatoria multidimensional uniforme cuyas componentes se distribuyen en el intervalo (0,1). La función

$$\Psi(\mathbf{Y}) = -D_{P_{\mathbf{Z}}|P_{\mathbf{U}}}$$

es una función contraste.

Además, si $\mathbf{q}(\cdot)$ es invertible, como la divergencia K-L es invariante frente a transformaciones invertibles, se tiene que la función contraste asociada al criterio INFOMAX o ENTMAX es

$$\Psi(\mathbf{Y}) = -D \left. p_{q^{-1}(z)} \right| p_{q^{-1}(u)} = -D \left. p_{\mathbf{Y}} \right| p_{q^{-1}(u)} = -D \left. p_{\mathbf{Y}} \right| \prod_{i=1}^N p_{q_i^{-1}(u_i)}$$

2.2.5 Relaciones entre criterios

Los criterios MMI, ML y ME están relacionados.

Supongamos que la f.d.p. de las fuentes es conocida

$$p_{\mathbf{S}} = \prod_{i=1}^N p_{S_i}$$

Entonces (puesto que $q^{-1}(U_i) = S_i$)

$$\Psi_{\text{ML}}(\mathbf{Y}) = -D \left. p_{\mathbf{Y}} \right| \prod_{i=1}^N p_{S_i} - h(\mathbf{X})$$

$$\Psi_{\text{ME}}(\mathbf{Y}) = -D \left. p_{\mathbf{Y}} \right| \prod_{i=1}^N p_{S_i}$$

Es decir, en BSS, donde $h(\mathbf{X})$ es constante para \mathbf{W} , los criterios de máxima verosimilitud y máxima entropía coinciden [Bell2000].

El criterio MMI es ligeramente distinto

$$\Psi(\mathbf{Y}) = -D \left. p_{\mathbf{Y}} \right| \prod_{i=1}^N p_{Y_i}$$

Sin embargo, localmente coinciden los tres.

2.2.6 Desajuste entre el modelo y la realidad

Los criterios ML e INFOMAX o ENTMAX requieren la estimación de las f.d.p. de las fuentes. Puesto que estas f.d.p. se desconocen se producirá un desajuste.

$$\Psi_{\text{ML}}(\mathbf{Y}) = -D \left. p_{\mathbf{Y}} \right| \prod_{i=1}^N p_{S_i} - h(\mathbf{X})$$

$$\Psi_{\text{ME}}(\mathbf{Y}) = -D \left. p_{\mathbf{Y}} \right| \prod_{i=1}^N p_{q_i^{-1}(u_i)}$$

Utilizando la propiedad de descomposición pitagórica de la divergencia K-L tenemos

$$\Psi_{ML}(\mathbf{Y}) = -D \prod_{i=1}^N p_{S_i} - D \prod_{i=1}^N p_{S_i} \left\| \prod_{i=1}^N p_{S_i} \right\| - h(\mathbf{X})$$

$$\Psi_{ME}(\mathbf{Y}) = -D \prod_{i=1}^N p_{S_i} - D \prod_{i=1}^N p_{S_i} \left\| \prod_{i=1}^N p_{q_i^{-1}}(U_i) \right\|$$

En ambos casos el primer término cuantifica la falta de separación mientras que el segundo cuantifica la diferencia entre la estima de las f.d.p. de las fuentes y las f.d.p. reales.

Cuando la estimación coincide con la realidad Ψ_{ML} y Ψ_{ME} son funciones contraste. En caso contrario, aunque la separación ($\mathbf{W} = \mathbf{A}^{-1}$) seguirá siendo un punto crítico de ambas funciones, ya no tiene porqué ser un máximo (puede ser un mínimo o un punto de inflexión).

2.3 EXTRACCIÓN CIEGA DE FUENTES

Hemos visto hasta ahora el problema de separación ciega de fuentes (BSS). Sin embargo, BSS puede consumir mucho tiempo de cálculo, sobre todo cuando el número de fuentes es muy elevado (esto ocurre, por ejemplo, en las aplicaciones biomédicas). En estos casos puede ser más deseable extraer sólo un conjunto de fuentes, y no todas. Esto se denomina extracción ciega de fuentes (**BSE**).

Pero, incluso si estamos interesados en separar todas las fuentes es más rápido utilizar BSE combinada con la deflación, consistente ésta en eliminar de las observaciones las componentes asociadas a las fuentes ya extraídas (las salidas resultantes de BSE), y volver a aplicar BSE para extraer nuevas fuentes. Este es un método alternativo (y más rápido) de llevar a cabo el proceso de BSS. Este método fue propuesto por primera vez por Delfosse y Loubaton y desarrollado por Hyvarinen y Oja y por Cichocky, Thawonmas y Amamri. Sin embargo, los algoritmos propuestos por ellos tenían la limitación de poder extraer las fuentes sólo de una en una, con lo que había que hacer una deflación por cada fuente extraída. Esto se debe a que de otra forma se producía una repetición de fuentes en las salidas (fuentes que aparecían en más de una salida, impidiendo así la extracción de otras).

En [Cruces2000] Cruces, Cichocki y Castedo consiguieron un algoritmo que combinaba BSE y deflación, extrayendo en cada paso un cierto número (mayor o igual

que 1) de fuentes. Este algoritmo resultó ser una generalización de del algoritmo FAST-ICA propuesto por Hyvarinen y Oja.

En muchas ocasiones lo que se pretende es extraer sólo un conjunto reducido de fuentes y no extraerlas todas (lo cual lleva mucho tiempo y la mayoría de las fuentes carece de interés). Sin embargo, el extraer un pequeño número de fuentes no nos asegura que esas fuentes sean las que nos interesa recuperar. Amari, Cichocki y yang propusieron un algoritmo que utilizaba como contraste la divergencia K-L entre la f.d.p. de las salidas y la f.d.p. (conocida a priori) de las fuentes deseadas. Esto tiene el inconveniente de que no siempre es conocida la f.d.p. de las fuentes deseadas. Por ello Cruces, Cichocki, Amari y Castedo propusieron en [Cruces2001] un algoritmo que extrae las fuentes con la kurtosis de mayor valor absoluto (es decir, las menos Gaussianas). Así se obtienen fuentes cuyo aspecto es menos ruidoso (tales como el latido del corazón, la interferencia del movimiento de los ojos, etc. en EEG o MEG).

2.3.1 Deflación

Supongamos que, mediante un algoritmo de extracción obtenemos como vector de salidas $\mathbf{y}_E = \mathbf{W}_E \mathbf{x}$, donde \mathbf{W}_E es la matriz de extracción y en vez de tener dimensión $N \times N$ (como ocurría con la matriz de separación \mathbf{W}) tiene dimensión $E \times N$, siendo E el número de fuentes extraídas. Es decir, el vector \mathbf{y}_E tendrá dimensión $E \times 1$.

Si \mathbf{A}_E es la pseudoinversa de \mathbf{W}_E entonces

$$\hat{\mathbf{x}} = \mathbf{A}_E \mathbf{y}_E$$

será la proyección de las fuentes extraídas sobre las observaciones \mathbf{x} . De este modo la deflación consiste en calcular

$$\mathbf{x}_D = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - \mathbf{A}_E \mathbf{B}_E \mathbf{x} = (\mathbf{I}_N - \mathbf{A}_E \mathbf{B}_E) \mathbf{x} = \mathbf{P} \mathbf{x}$$

\mathbf{I}_N es la matriz identidad de orden N , y \mathbf{x}_D es el nuevo vector de observaciones para el siguiente paso de BSE, en el que ya no estarán presentes las fuentes extraídas.

2.4 CONCLUSIONES

Hemos estudiado en este capítulo el problema de BSS para mezclas lineales instantáneas. Hemos visto que, cuando no más de una fuente es Gaussiana, la separación de las fuentes se producirá cuando las salidas sean mutuamente

independientes. Además, en mezclas lineales, esta condición se puede relajar y sustituir por la independencia dos a dos de las salidas.

Hemos visto que también que la independencia de las fuentes puede ser comprobada mediante la utilización de momentos y cumulantes de orden superior. Esto permite trabajar en condiciones de ruido aditivo Gaussiano, debido a que sus estadísticos de orden superior son nulos.

Hemos estudiado el enfoque de análisis de componentes independientes (ICA) y la utilización del análisis de componentes principales (PCA), tanto para realizar el preblanqueo (decorrelación) de las observaciones como para reducir su número manteniendo las componentes más importantes.

Hemos comprobado también cómo los criterios de separación pueden ser estudiados de forma general utilizando la Teoría de la Información, y cómo los criterios MMI, ML y ME se relacionan mediante la divergencia KL entre la f.d.p. conjunta de las salidas y el producto de f.d.p. marginales (ya sean las salidas o las fuentes estimadas). Sin embargo, cuando la f.d.p. de las fuentes es desconocida el único criterio de los tres que garantiza la separación es el criterio MMI.

Finalmente hemos visto un método alternativo a la BSS: la extracción ciega de fuentes (BSE), ya sea para extraer un reducido grupo de fuentes de interés, ya sea para combinarlo con la deflación y extraer por grupos todas las fuentes. Este método es más rápido que la BSS.

CAPÍTULO 3

APLICACIÓN DEL ICA A LAS SEÑALES BIOLÓGICAS

En el campo de la ingeniería biomédica existen numerosas aplicaciones del ICA. Las señales biomédicas son medidas de diferentes características (eléctricas, magnéticas, hemodinámicas) de partes cuerpo humano recogidas simultáneamente por una cierta cantidad de sensores situados en posiciones diferentes, suficientemente alejados para cubrir una gran área, pero suficientemente cercanos para que no queden lagunas sin cubrir.

Esto da lugar a dos problemas fundamentales. En primer lugar, los diferentes sensores van a recoger tanto señales de interés para el estudio que se pretenda llevar a cabo como ruido y otras señales producidas por el cuerpo que no tienen interés y que interfieren en las señales que sí lo tienen. En cada sensor aparecerán mezcladas distintas componentes. Esas señales indeseadas que se superponen a las señales de interés se denominan artefactos, y pueden llegar a ser un obstáculo para el médico que debe analizar las medidas. Uno de los objetivos del ICA en su aplicación a las señales biomédicas es la eliminación de los artefactos (separando las componentes independientes y luego haciendo deflación de los artefactos).

Por otra parte, la distancia que separa los sensores de las fuentes que producen las propias señales de interés hace que éstas también aparezcan mezcladas entre sí en los sensores. Cada fuente aparecerá superpuesta a otras en varios sensores (o en todos). Esto hace que las señales de mayor nivel enmascaren a las más débiles. Además, las señales de niveles parecidos darán lugar a formas de onda en las que será imposible reconocer a ninguna de ellas. Por lo tanto, otro de los objetivos de la aplicación del ICA

a las señales biomédicas es la separación de las fuentes objeto de estudio, haciendo posible el reconocimiento de cada una de las fuentes de interés.

El ICA tiene otra utilidad más: una vez que ha sido llevada a cabo la separación es posible proyectar cada componente independiente sobre los sensores, de manera que, tanto los artefactos como las fuentes pueden ser localizados. En el caso de los artefactos se localiza la fuente si el artefacto está originado en la zona donde se toman las medidas. En caso contrario lo que se localiza es la zona donde el artefacto se manifiesta con más fuerza. En el caso de las señales de interés la proyección de las componentes en los sensores es de gran utilidad, ya que permite localizar el origen de distintos fenómenos, lesiones o enfermedades, conocer mejor la actividad neuronal, etc.

Ejemplos de medidas biomédicas son el electroencefalograma (EEG), el magnetoencefalograma (MEG), la resonancia magnética funcional (fMRI), el electrocardiograma (ECG) o los electrooculogramas vertical (VEOG) y horizontal (HEOG), entre otras.

3.1 APLICACIÓN DEL ICA AL ECG

El ECG es una técnica para la medida de las señales eléctricas emitidas por el corazón en su funcionamiento. Se realiza con varios sensores cercanos entre sí situados en el pecho y el abdomen del paciente. Las partes del corazón (aurículas, ventrículos, etc.) dan lugar a componentes eléctricas independientes (sobre esto existen discusiones) y con patrones temporales y espaciales diferentes. Esto quiere decir que la señal recogida por cada uno de los sensores será una mezcla de las señales emitidas por cada una de las componentes del corazón. Por lo tanto el análisis de componentes independientes aparece como una herramienta que puede permitir al médico cardiólogo estudiar por separado con mayor fiabilidad el comportamiento de las distintas partes del corazón, es decir su funcionamiento interno, de manera que se facilita también el diagnóstico de enfermedades cardiovasculares, lesiones, etc.

Según [Jung2000] los ECG satisfacen las siguientes condiciones:

1. las mezclas de las componentes que dan lugar a las señales recibidas en los sensores son lineales

2. los tiempos de retraso en llegar estas señales a la superficie torácica y abdominal (y, por lo tanto, a los sensores) son despreciables, por lo que la mezclas se pueden considerar instantáneas
3. en principio, parece ser que el número de fuentes es menor que el de sensores
4. la distribución de las fuentes es no Gaussiana.

Respecto al punto (3) hay que decir, sin embargo, que los movimientos (como las contracciones) producidos en el corazón como consecuencia de su actividad impiden la estacionariedad espacial de las fuentes, por lo que habrá que considerar que existen múltiples fuentes independientes de duración limitada en el tiempo asociadas a cada fuente de actividad eléctrica del corazón. Esto significa que el problema de descomponer las distintas fuentes internas del corazón no es tan sencillo como parece en un principio.

Otra discusión existente es acerca de la independencia de esas componentes asociadas. Según [Jung2000], puesto que las señales producidas por esas fuentes están generadas en diferentes instantes de tiempo, para los algoritmos de ICA son independientes.

3.1.1 Extracción del ECG del feto

La aplicación más usual y más estudiada del ICA con respecto al ECG es la obtención del ECG del feto (FECG) de una mujer embarazada mediante una técnica no invasiva [DeLathauwer2000]. El FECG contiene información importante acerca de la salud del feto, y su forma de onda, así como su pulso cardíaco (FHR) pueden revelar diversos problemas, como arritmias, etc.

Es interesante, pues obtener el FECG. Sin embargo lo que se pretende es obtenerlo desde el exterior, evitando con ello perturbar las condiciones en que se encuentra el feto.

La técnica utilizada para ello consiste en colocar electrodos como los utilizados en la obtención del ECG en el vientre de la madre, donde con más fuerza se podrá mostrar el latido del feto. El problema que esto plantea es que el latido de la madre tiene mucha más fuerza que el del feto, por lo que en los sensores aparecerán superpuestos el

ECG materno (MECG) y el del feto, siendo además dominante el MECG, de modo que distinguir el FECG se hace muy difícil (no tanto la frecuencia de pulso como la forma de onda, de la que poco se puede deducir).

Para resolver este problema se utiliza el análisis ICA, separando el MECG del FECG.

Otras fuentes aparecen mezcladas en los sensores, como la actividad eléctrica generada por la respiración y el estómago de la madre, la actividad electromiográfica generada por los músculos, el útero, etc., así como el ruido (el térmico, el provocado por el contacto de los electrodos con el vientre, etc.) o la interferencia provocada por la red eléctrica de 50 Hz.

Todas estas fuentes pueden ser importantes debido al bajo voltaje producido por el latido tan débil del corazón del feto, y también el análisis ICA las separará del FECG (al menos aquellas que no sean Gaussianas). No obstante el artefacto más importante con que nos encontramos cuando aplicamos esta técnica no invasiva para obtener el FECG es el MECG.

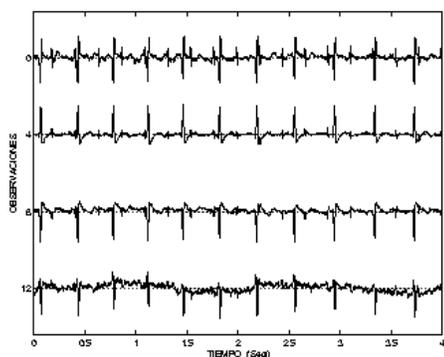


Figura 3-1 Observaciones de un FECG

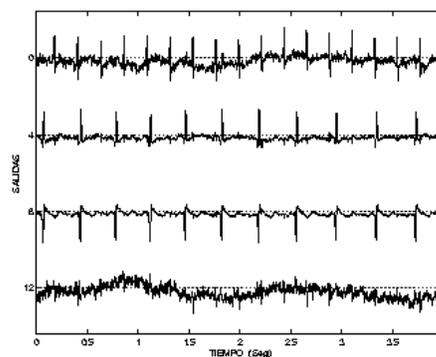


Figura 3-2 Componentes tras análisis ICA

En la figura 3-1 se muestra un conjunto de 4 observaciones en el que el FECG aparece muy oculto por el MECG. En la figura 3-2 se muestran los resultados de realizar ICA sobre estas observaciones. La primera componente es el FECG, las dos siguientes el MECG y la última es ruido generado por otras actividades.

Según [DeLathauwer2000], a una cierta distancia del corazón de la madre, su actividad eléctrica se puede modelar como un vector de señal tridimensional, que sería el efecto de un dipolo de corriente que giraría en el pecho de la madre. Este vector

tridimensional, cuando se muestrea (y se convierte en una matriz $3 \times N$), es lo que en [DeLathauwer2000] se denomina subespacio MECG.

Por el contrario la dimensión del correspondiente subespacio FECG (el número de señales independientes que describen la actividad eléctrica del corazón del feto) es desconocido y cambiante a lo largo del embarazo.

Por otra parte, el canal que une las fuentes con los electrodos puede considerarse lineal y resistivo, y las señales de baja frecuencia en comparación con su tiempo de propagación para llegar a los electrodos. Por lo tanto se pueden asumir las hipótesis acerca de que las mezclas son lineales e instantáneas.

En [DeLathauwer2000] se propone la utilización de un método denominado BSS (separación ciega de subespacios de fuentes), consistente en separar grupos de señales (subespacios) para (en el caso concreto de la búsqueda del FECG) separar el subespacio MECG del subespacio FECG.

Tanto el método citado como otros métodos utilizados (por ejemplo, el INFOMAX extendido utilizado en [Jung2000]) muestran la existencia de varias componentes independientes en el FECG y en el MECG.

3.1.2 BSS de parámetros del ECG

Otra aplicación del ICA referente al ECG es la separación de componentes independientes a partir, no de las observaciones procedentes del ECG sino de parámetros del mismo.

Un ejemplo de ello lo tenemos en [Virag2000]. Es conocida la relación existente entre las alteraciones del reflejo autónomo y diversas patologías fisiológicas. Por ello es importante el estudio de la modulación, latido a latido, del reflejo cardiaco autónomo.

El análisis espectral de la señal cardiaca procedente del ECG revela que las componentes de alta frecuencia (aquellas que están en la banda 0.15 - 0.4 Hz) están relacionadas con la modulación parasimpática, mientras que no queda claro el papel de las componentes de baja frecuencia (0.04 - 0.15 Hz).

Debido a esto el propio Virag ya propuso anteriormente un método basado en BSS que separa las componentes de baja frecuencia de la evolución en el tiempo de dos parámetros. Uno de ellos, el ritmo cardiaco (RR) procedente del ECG. El otro, la

presión arterial (ABP), relacionado con el latido cardiaco, aunque no presente en el ECG. Este método se mostró válido para la observación de los cambios en el flujo cardiaco autónomo.

El método citado tiene sin embargo el inconveniente de la necesidad de realizar simultáneamente en el paciente la medida de la presión arterial y el ECG, lo cual es bastante engorroso e incómodo.

En [Virag2000] se propone un nuevo método que analiza las fluctuaciones a corto plazo producidas en la evolución temporal del ritmo cardiaco (RR) y el intervalo QT. La señal de un electrocardiograma tiene varios puntos significativos: el punto P es un pequeño pico. Tras él está el punto Q, que es una pequeña bajada tras la cual inmediatamente se encuentra el gran pico que es el punto R. Tras él hay una bajada algo mayor que la producida en el punto Q, y se trata del punto S. A la subida posterior se le denomina punto T. Es por esto que al ritmo cardiaco se le denomina RR, ya que es el intervalo correspondiente a un período. El intervalo QT, obviamente, es el intervalo existente entre el punto Q y el T.

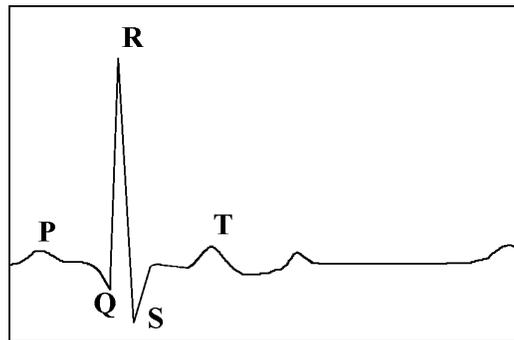


Figura 3-3 Puntos significativos del ECG

La ventaja del método propuesto en [Virag2000] es que sólo precisa del ECG, prescindiendo de la medida de la presión arterial. Este método se basa en conocimiento del control que sobre el corazón ejercen las dos partes antagónicas del sistema nervioso autónomo (ANS): la actividad cardiaca simpática (CSNA) y la parasimpática (CPNA). Este control queda reflejado en la evolución de parámetros como RR y QT.

También la respiración tiene influencia sobre los parámetros del ECG. Sin embargo la influencia importante se produce a través del propio sistema nervioso

autónomo, siendo despreciable la contribución mecánica de la respiración en dichos parámetros.

El método, que pretende observar el flujo cardíaco autónomo, realiza BSS a las señales observadas RR y QT para obtener por separado las fuentes (las señales CSNA y CPNA). Esto, sin embargo, implica considerar que esas fuentes CSNA y CPNA son independientes, lo cual no es cierto, por lo que el algoritmo obtiene en realidad dos señales independientes que tienen, respectivamente, una gran influencia en CSNA y en CPNA (esta hipótesis se basa en trabajos previos de los autores de [Virag2000]).

Las otras dos hipótesis para poder aplicar BSS son:

1. considerar las fuentes lineales
2. considerar el ruido como Gaussiano y blanco.

En resumen: es posible estudiar el comportamiento del flujo cardíaco autónomo y las patologías relacionadas a partir de la aplicación del ICA o la BSS a la evolución en el tiempo de parámetros del ECG.

3.2 ICA APLICADO A fMRI

La obtención de imágenes por resonancia magnética (MRI) es una técnica utilizada para conseguir imágenes de diversas partes del cuerpo sin necesidad de emplear rayos X.

La técnica consiste en situar al paciente dentro de un amplio y fuerte campo magnético. Mediante una antena de radio se envían señales al paciente y se reciben de vuelta por esa misma antena. Prácticamente es posible tomar imágenes desde cualquier ángulo.

La resonancia magnética funcional (fMRI) consiste en la utilización de la MRI, junto con diversas tareas relacionadas con el aparato psicomotor que el paciente debe realizar, para el estudio de la actividad cerebral durante estas tareas.

El funcionamiento de la fMRI [Lee98] se basa en la diferente susceptibilidad paramagnética que poseen la oxihemoglobina y la desoxihemoglobina, lo cual hace que un cambio en el nivel de oxígeno en la sangre sea detectado por el escáner de fMRI.

Esta técnica utilizada por la fMRI se denomina contraste dependiente del nivel de oxígeno en la sangre (BOLD: Blood Oxygenation Level Dependent).

El problema que presenta la fMRI es que perturbaciones como el ruido de la máquina, pequeños movimientos del paciente, el latido cardiaco o su respiración no permiten detectar correctamente la actividad cerebral relacionada con las tareas de prueba (actividad reflejada únicamente en pequeños cambios locales en el nivel de oxígeno). Es decir, los artefactos tienen un nivel tal que se hace muy difícil, o imposible, el análisis de los datos obtenidos.

Por ello, diversos autores proponen la utilización del ICA sobre los datos de fMRI. Para una misma posición se toman N imágenes (N ~ 100) de T puntos cada una (T ~ 10000).

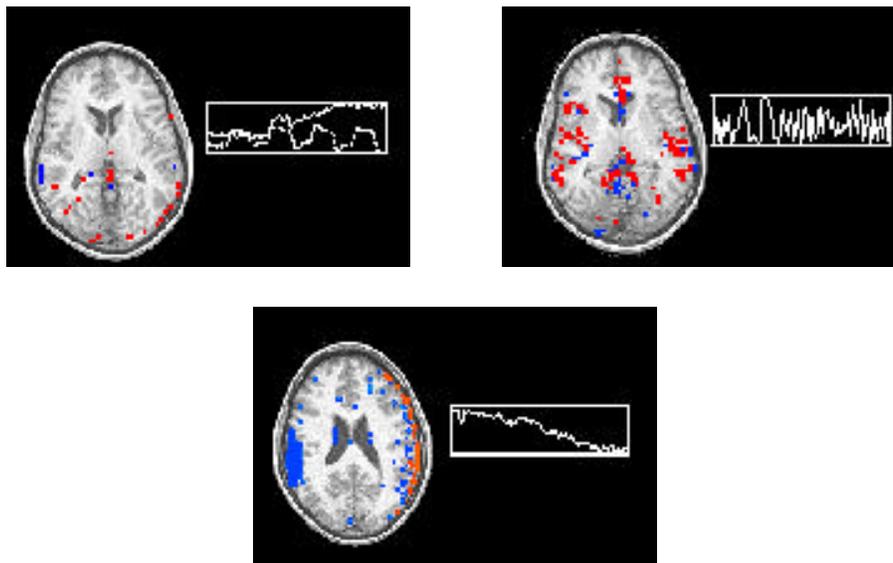


Figura 3-4 Algunas componentes que se pueden encontrar en las imágenes de fMRI (el color rojo indica zona activa y el azul zona inactiva): componente transitoria asociada a tarea (arriba a la izquierda), componente cuasiperiódica (arriba a la derecha) y componente debida al movimiento lento de la cabeza (abajo)

En [Lee98] se muestran dos maneras diferentes de resolver el problema aplicando ICA.

La primera de ellas consiste en aplicar ICA considerando que las observaciones proceden de N sensores (cada imagen es considerada como un sensor) y cada sensor

recoge T instantes distintos (cada punto es considerado como un instante). De este modo, la matriz de separación \mathbf{W} es de orden $N \times N$. Se consiguen así mapas independientes como resultado.

La otra manera de resolver el problema es considerar que existen T sensores (cada uno de los puntos de la imagen) que recogen señales de N muestras cada uno (cada imagen será un instante de tiempo). Por lo tanto \mathbf{W} es de orden $T \times T$. Esto da lugar a cálculos muy complejos que ralentizan el proceso enormemente (hay que recordar que T es del orden de 10000).

Para evitarlo se definen regiones de interés (ROI), que son aquellos puntos cuya función en el tiempo está altamente correlada con la función de referencia (según la tarea en cuestión). Posteriormente se aplica PCA para reducir aún más la dimensión.

Ambos métodos proporcionan resultados satisfactorios a la hora de localizar la actividad cerebral relacionada con distintas tareas.

3.3 APLICACIÓN DE ICA A EEG Y MEG

Una de las más importantes aplicaciones del ICA en el campo de la medicina es el análisis de los datos obtenidos en un electroencefalograma (EEG) o en un magnetoencefalograma (MEG).

Ambas técnicas proporcionan información acerca de los campos eléctricos y magnéticos provocados por las corrientes neuronales del cerebro y propagados hacia la superficie de la cabeza [Vigário2000].

El EEG se lleva a cabo mediante la colocación de electrodos en diversos puntos de la superficie de la cabeza y que miden la diferencia de potencial entre pares de puntos, del mismo modo que el ECG realiza esta medida en puntos cercanos al corazón [Vigário2000, Lee98].

La técnica empleada por el MEG consiste en realizar medidas del flujo magnético que atraviesa un área en diversos puntos exteriores a la superficie de la cabeza. Esta medida se lleva a cabo con los denominados SQUIDS (superconducting quantum interference devices). Estos dispositivos tienen todos una misma superficie, y miden el flujo magnético ortogonal que la atraviesa [Hari97].

En cada punto a medir se sitúan dos SQUIDs ortogonales, de manera que, multiplicando los flujos por las superficies (operación innecesaria, ya que todas las superficies son iguales) tenemos en cada punto dos magnitudes escalares medidas. Puesto que se trata de magnitudes escalares, por cada punto tendremos dos observaciones.

Por lo tanto, si el número de puntos de la cabeza donde hay SQUIDs es $N/2$ tenemos N observaciones.

Las formas de onda (por ejemplo, el latido del corazón, que se manifiesta tanto en el EEG como en el MEG) recogidas por el MEG son iguales que las provenientes de las mismas fuentes recogidas por el EEG.

Los EEG y los MEG son utilizados por los médicos para estudiar el comportamiento cerebral, la actividad neuronal en diversas situaciones. Son útiles para el diagnóstico de enfermedades, lesiones, mal funcionamiento o bien para el estudio de las reacciones cerebrales ante diversos estímulos (auditivos, visuales, táctiles, etc.), lo cual arroja luz sobre el funcionamiento del cerebro, un órgano poco conocido aún [Vigário2000, Lee98].

Un problema que surge cuando se recogen datos utilizando las técnicas del EEG o del MEG es que no sólo aparecen registradas las señales de interés (las originadas por la actividad cerebral en situación normal o la provocada por los estímulos presentados al paciente), sino que también aparecen otras señales indeseadas, fruto de otras actividades realizadas en la cabeza (movimiento de los ojos, parpadeos, movimiento de las mandíbulas, tensión de diversos músculos faciales, etc.), fuera de ella (latido del corazón, movimiento del pecho debido a la respiración, etc.) o incluso fuera del cuerpo (ruidos electromagnéticos provocados por las propias máquinas, por la presencia de aparatos electrónicos como relojes, etc.). Es decir, se recogen artefactos.

A veces los artefactos pueden tener una gran amplitud y enmascarar intervalos interesantes de los datos (como ocurre con el caso de los parpadeos y movimientos de ojos). Otras veces, incluso, su forma de onda puede confundirse con señales procedentes de la actividad cerebral y dar lugar a diagnósticos o conclusiones erróneos [Lee98].

Para eliminar los artefactos, como ya hemos visto en otros tipos de medidas, se utiliza el análisis ICA y la posterior deflación de las componentes identificadas como artefactos, de manera que los datos quedan limpios de los mismos para el posterior análisis médico.

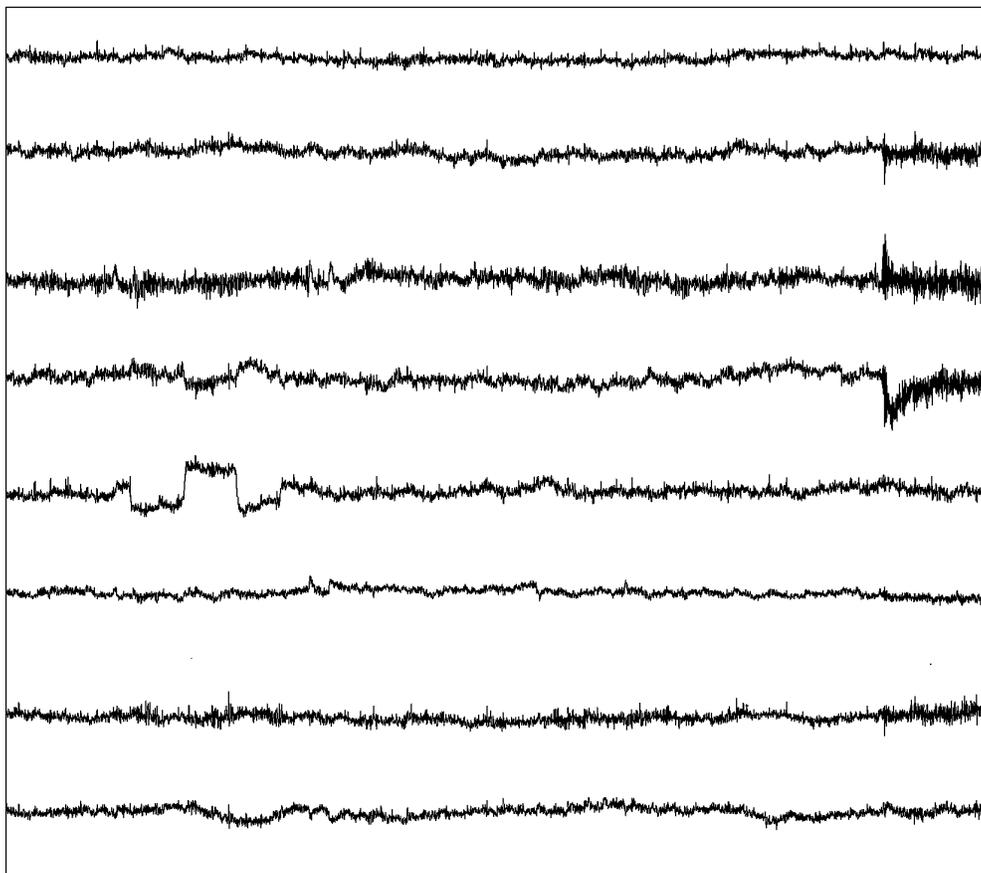


Figura 3-5 Artefactos en observaciones de un MEG

En la figura 3-5 vemos las observaciones en un conjunto de sensores de un neuromagnetómetro. Se puede comprobar la presencia del artefacto correspondiente a un movimiento horizontal de ojos (sobre todo en la quinta observación), del correspondiente al latido cardíaco (muy fuerte en la primera y en la última observaciones) o del correspondiente a la acción de morder (muy apreciable en la segunda, tercera y cuarta observaciones).

Sin embargo, además de los artefactos existe otro inconveniente que se presenta siempre que se realizan medidas con varios sensores a cierta distancia de las fuentes: las señales recogidas por el EEG o el MEG son mezclas de las señales originadas en las distintas partes del cerebro y que constituyen las verdaderas señales de interés.

Por lo tanto, se desea también poder separar las señales de interés antes del estudio médico para así poder examinarlas por separado, evitando además que unas señales interfieran en otras.

El ICA se muestra como una buena herramienta para la separación de señales producidas por la actividad cerebral en componentes independientes. De esta manera es posible realizar experimentos en los que más de un estímulo se presenta al paciente, pudiéndose realizar por separado el estudio de las reacciones a cada uno de esos estímulos.

Otra utilidad de la aplicación del ICA a los datos de un EEG o un MEG es la posibilidad de proyectar las componentes obtenidas del análisis sobre los sensores, de modo que se pueden localizar las fuentes de esas componentes. Es decir, es posible localizar en el cerebro las fuentes de los artefactos (o al menos, los lugares de la cabeza donde estos artefactos se muestran con más intensidad) y las partes del cerebro que originan los diversos fenómenos (respuestas a los estímulos, actividad habitual) [Vigário2000, Barros2000, Jung2000, Lee98].

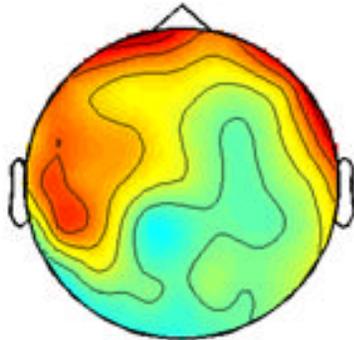


Figura 3-6 Mapa de la componente debida al latido cardiaco

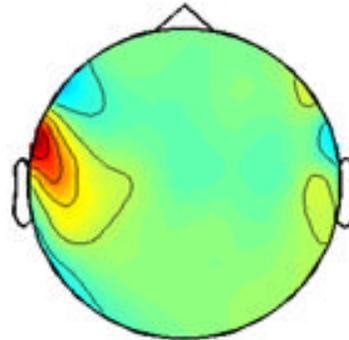


Figura 3-7 Mapa de la componente debida a la mordedura

En las figuras 3-6 y 37 vemos dos ejemplos de mapas cerebrales realizados por la proyección de dos componentes: el latido del corazón y la actividad muscular debida a la acción de apretar las mandíbulas. La forma del mapa ayuda en muchas ocasiones a identificar las componentes.

Veamos: si $\mathbf{y} = \mathbf{W}\mathbf{x}$ son las componentes independientes surgidas del análisis ICA, entonces las observaciones son $\mathbf{x} = \mathbf{W}^{-1}\mathbf{y}$, de manera que la columna i -ésima de \mathbf{W}^{-1} es la proyección de la la componente i -ésima sobre los sensores. La zona donde

aparezca un mayor nivel al hacer dicha proyección será la zona donde se encuentra el origen de esa componente.

Las señales originadas por estímulos son denominadas event-related signals o event-related potentials (ERP) [Vigário2000, Barros2000, Jung2000, Lee98].

Vemos pues que el ICA es útil tanto para eliminar los artefactos de los EEG y los MEG como para el propio análisis de los datos de los EEG y los MEG.

La aplicación del ICA con éxito a los datos de los EEG y los MEG es posible porque se cumplen una serie de condiciones [Vigário2000, Jung2000, Lee98]:

1. fuentes independientes
2. mezclas instantáneas y lineales
3. fuentes no Gaussianas
4. proceso de mezcla estacionario.

La independencia de las fuentes es una independencia temporal de las señales originadas en esas fuentes, lo cual no tiene que suponer forzosamente independencia en las estructuras fisiológicas de esas fuentes (aunque, en el caso de los artefactos, fue esta independencia fisiológica la que llevó a suponer la independencia estadística de las señales [Vigário2000]).

En muchos algoritmos se sustituye la independencia por la búsqueda de la no gaussianidad, debido a que tanto los artefactos como las señales neuronales del cerebro son no Gaussianas. Sin embargo, la actividad cerebral consiste en muchas ráfagas de señales oscilatorias no Gaussianas, pero que, dependiendo de la longitud de esas ráfagas, pueden tener kurtosis negativa, positiva o nula. En este último caso se confunden con señales Gaussianas, por lo que estos algoritmos que buscan la no gaussianidad tendrán que utilizar criterios adicionales.

Las mezclas se pueden considerar instantáneas debido a que las señales son de frecuencias lo suficientemente bajas como para que los tiempos de propagación de las mismas desde sus orígenes hasta los sensores sean despreciables.

Las señales originadas por las fuentes no son realmente estacionarias, pero no importa, porque los algoritmos tratan los datos como variables aleatorias y no las fuentes como procesos estocásticos.

El considerar que el proceso de mezcla es estacionario significa asumir que las fuentes permanecen inmóviles y sin cambios en su orientación. Ello implica considerar la actividad cerebral como un conjunto de fuentes inmóviles y aisladas, en vez de cómo un conjunto de ondas que se desplazan. Los resultados que se han obtenido en los diferentes estudios que se han venido realizando a lo largo de estos últimos años indican que es posible realizar esta suposición, la cual conlleva la suposición de la matriz de mezclas (y, por lo tanto, la matriz de separación) como una matriz de coeficientes constantes.

Como siempre, el ICA sólo puede separar un número de fuentes menor o igual al de sensores. Según [Lee98], a pesar de que el número de señales independientes presentes en el cerebro se desconoce, los resultados muestran que el ICA es efectivo separando señales importantes en presencia de gran número de señales independientes con niveles de amplitudes pequeños.

3.3.1 MEG

El estudio del funcionamiento cerebral a través de los MEG está menos extendido que a través de los EEG. También la aplicación del ICA a los datos recogidos por los MEG está menos desarrollada.

No obstante, en [Vigário97] se mostró que el ICA se puede aplicar a los datos de MEG con igual éxito que a los de EEG, comprobándose con la aplicación del algoritmo FAST-ICA de Hyvärinen y Oja de 1997.

Como ya se ha dicho, las formas de onda recogidas por un neuromagnetómetro (aparato con el cual se lleva a cabo la realización de un MEG) son iguales que las recogidas por los electrodos con que se realiza un EEG. Lo cual quiere decir que podemos aplicar el ICA a los datos de un MEG con el mismo resultado que a los de un EEG.

Sin embargo la técnica MEG tiene algunas ventajas sobre el EEG. En un EEG los electrodos, en contacto directo material con la cabeza del paciente, introducen una alteración (un ruido eléctrico) que se refleja en las medidas recogidas. En cambio, los SQUIDS del MEG no están en contacto con la cabeza, sino que recogen el campo magnético que sale de ella.

El MEG aprovecha el hecho de que tanto el cuero cabelludo como el cráneo (ambos tejidos no homogéneos) son transparentes a los campos magnéticos [Hari97]. De modo que los campos magnéticos no sólo llegan a los SQUIDS sin sufrir modificaciones al pasar por el cuero cabelludo, sino que llegan a este sin perturbaciones provocadas por el cráneo. Es decir, los SQUIDS recogen los campos magnéticos tal como salen de la corteza cerebral.

El MEG es, pues, una técnica no invasiva que permite observar el flujo magnético en la corteza cerebral de un ser vivo, sin que el cráneo o el cuero cabelludo sean un obstáculo para ello.

Una técnica alternativa sería la invasiva con microelectrodos, que tiene la ventaja de poder identificar fácilmente las fuentes, pero que tiene la desventaja (con respecto a MEG y a EEG) de no poder obtener datos de amplias regiones de la corteza cerebral y de ser invasiva.

3.3.2 Eliminación de artefactos

La eliminación de artefactos es un paso importante cuando se pretende analizar los datos de un EEG o de un MEG.

Artefactos como los provocados por el movimiento de los ojos, el parpadeo, el movimiento de los músculos faciales, el latido del corazón, etc. pueden ser lo suficientemente elevados como para ocultar partes importantes de las señales procedentes de la actividad cerebral.

Antes de la aplicación del ICA el método utilizado para eliminar los artefactos consistía en ignorar las partes del EEG o el MEG donde el nivel de los artefactos sobrepasaba un determinado umbral [Lee98]. Pero esto tiene dos inconvenientes. Por un lado se corre el riesgo de eliminar una parte muy importante de la información (oculta bajo los artefactos). Por otra parte los artefactos pueden ocupar una gran extensión en el tiempo, ya sea debido a la naturaleza de los mismos (como ocurre en el caso del latido cardiaco), ya sea por la imposibilidad de controlar al paciente (enfermos mentales o, simplemente, personas nerviosas incapaces de permanecer completamente quietas). Evidentemente, esta clase de artefactos no pueden ser eliminados ignorando las zonas de los datos en que se manifiestan porque desaparecería la mayor parte de los datos.

Realizar el análisis ICA a los datos de un EEG o un MEG permite separar los artefactos (y eliminarlos) y proyectarlos sobre los sensores. Esto último tiene la utilidad de proporcionar otro medio de asegurar que se trata de artefactos. Además de su forma de onda se contará con su localización. Por ejemplo, los parpadeos tendrán una mayor proyección en la zona más cercana a los ojos. Otro ejemplo es el latido cardiaco, que se manifiesta, según [Vigário97] y según veremos en el capítulo de simulaciones, en el lado izquierdo de la cabeza más que en el derecho.

3.3.3 ERP

Una parte de la actividad cerebral consiste en la respuesta ante estímulos, y a partir de un EEG o de un MEG es posible estudiar este aspecto del comportamiento del cerebro.

Cuando el cerebro recibe un estímulo interno o externo (auditivo, visual, táctil, etc.) genera una respuesta a ese estímulo. Esa respuesta está limitada en el tiempo por el estímulo, de manera que se dice que es una señal relacionada con un evento (event-related signal). A los potenciales así generados se les llama event-related potentials (ERPs) [Vigário2000, Barros2000, Jung2000].

Tradicionalmente se recurre al promediado de varias pruebas para aumentar la relación señal a ruido de los ERP (considerando como ruido tanto los artefactos como la actividad habitual del cerebro y otros ruidos electromagnéticos).

Si embargo los resultados no son satisfactorios con frecuencia, debido a la variación en cuanto a la posición en el tiempo de las respuestas y a su distribución en los sensores [Barros2000, Jung2000].

Por eso se ha comenzado a estudiar la utilización del ICA para la extracción de los ERPs de entre las observaciones, y también para la separación de ERPs debidos a estímulos diferentes.

Los problemas con que nos encontramos al intentar analizar los ERPs presentes en un EEG o en un MEG y que se pretenden resolver con la aplicación del análisis ICA son:

- Presencia de artefactos que deben ser identificados y eliminados

- Actividad cerebral "de fondo" (es decir, la actividad habitual cerebral) que suele tener un nivel mayor que los ERPs, por lo que la relación señal a ruido de éstos son muy bajas
- Variación entre pruebas de las amplitudes y las posiciones en el tiempo tanto de los ERPs como de la actividad continua del cerebro.

Con respecto a la estacionariedad espacial exigida por el ICA para su aplicación hay que decir que, a pesar de que algunos fenómenos espontáneos concretos (spindles, algunas depresiones, etc.) plantean problemas, la mayoría de las señales de un EEG o un MEG (incluidos los ERPs) cumplen este requisito de estacionariedad espacial [Lee98].

La hipótesis de independencia de las fuentes ha de ser examinada en los casos particulares en que se estudian varios ERPs simultáneamente (es decir, con distintas pruebas simultáneas), ya que, a veces, ERPs debidos a estímulos distintos son, en realidad, sumas de perturbaciones en las componentes habituales de del EEG o el MEG [Lee98], y, por lo tanto, no son totalmente independientes, por lo que el análisis de componentes independientes no va a proporcionar como resultados los ERPs, sino las componentes de éstos.

Diversos algoritmos han sido propuestos para la superación de todos estos obstáculos y el análisis ICA se revela como una buena herramienta para el estudio de los ERPs.

Numerosos experimentos proporcionan buenos resultados, como vemos en [Jung2000] (detección de señal tipo alfa, monitorización de alerta) o en [Vigário2000] (separación de ERPs de estímulos auditivo y táctil).

A pesar de ello hay que examinar los casos particulares, ya que, en general, no puede decirse que las señales ERP cumplan las hipótesis necesarias para realizar el análisis ICA.

Algunas de las oscilaciones electromagnéticas más conocidas y que aparecen al aplicar el ICA a los EEG o los MEG son:

- Ritmo alfa: sus fuentes se encuentran en las zonas posteriores del cerebro y están relacionadas con la movilidad de los ojos (no se debe confundir este tipo de señal con el artefacto provocado por los diversos músculos que mueven los ojos y que se localiza, como ocurre con los parpadeos, en la parte

delantera del mapa cerebral, ya que se genera en los propios ojos). Tiene un pico alrededor de 10 Hz.

- Ritmo mu: sus fuentes se hallan en las regiones centrales del cerebro, y se relacionan con la movilidad de las extremidades y con los estímulos táctiles. Tiene un pico alrededor de 10 Hz y otro alrededor de 20 Hz.
- Ritmo tau: sus fuentes están en la región supratemporal, y se relacionan con los estímulos auditivos y el sueño.

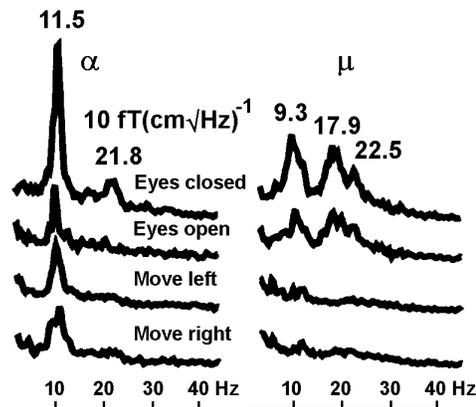


Figura 3-8 Espectros de frecuencia de de los ritmos alfa y mu

En la figura 3-8 se observan los espectros de frecuencia de los ritmos alfa y mu para distintas acciones del paciente: ojos cerrados, ojos abiertos, movimiento de la mano izquierda y movimiento de la derecha.

3.3.4 Algunas aplicaciones concretas aún no resueltas

Veremos ahora dos aplicaciones del análisis ICA que, proporcionando resultados alentadores, plantean algunos problemas que deberán ser resueltos en un futuro.

- **Identificación de tumores cerebrales:**

Una aplicación concreta del análisis ICA de los datos de un EEG es la identificación de tumores cerebrales, según los experimentos mostrados en [Bauer2000].

En dichos experimentos, en los que se utiliza un paciente con un meningioma canceroso localizado en la zona frontal izquierda del cerebro, aparecen dos componentes que no aparecen en los EEG realizados en pacientes normales.

Esas componentes tienen formas de onda denominadas ondas delta, y su frecuencia característica es el intervalo 1.5 - 3-5 Hz.



Figura 3-9 Ondas delta

En la figura 3-9 se muestran dos componentes delta encontradas en [Bauer2000] tras el análisis ICA de un paciente con tumor cerebral.

También se presenta en [Bauer2000] un método de identificación automática de las distintas componentes de surgidas del análisis ICA.

El problema que aparece es que de las dos componentes encontradas que parecen relacionarse con el tumor (debido a que no aparecen en pacientes sanos) sólo una de ellas, en su proyección sobre los sensores, se localiza en la zona donde se encuentra el tumor.

Los resultados, pues, no son completamente exitosos, por lo que es necesario desarrollar nuevos algoritmos que permitan la identificación y localización de los tumores mediante el análisis de los datos de EEG y MEG, sin necesidad de recurrir a las radiaciones ionizantes utilizadas en radiología.

- **Detección de spindles:**

Los spindles son señales que aparecen en los EEG y los MEG realizados en pacientes que están dormidos. Cuando más aparecen es en la segunda etapa del sueño.

Se trata de señales cuya banda de frecuencias es bastante ancha (de 11.5 a 16 Hz).

La frecuencia de aparición de los spindles es uno de los criterios utilizados para establecer una clasificación de las etapas del sueño. Por lo tanto, una

correcta detección automática de los spindles sería muy útil para mejorar la precisión de esta clasificación.

Tradicionalmente se ha llevado a cabo la detección visual de los spindles, lo cual lleva mucho tiempo en EEG multicanales (con muchos sensores), además de ser posible la aparición de artefactos.



Figura 3-10 Señal con varios spindles

En 1994 Schmicek propuso un algoritmo basado en la frecuencia, amplitud y duración de los spindles para la detección automática de éstos. Sin embargo, aparecían problemas en EEG multicanales.

En [Rosipal98] se propone y estudia el uso del ICA para separar un spindle, que será utilizado para clasificar sucesivos spindles.

El método funciona bien con una mezcla artificial de un spindle, una señal periódica y una Gaussiana. Pero con datos reales surgen dificultades (falsas detecciones) debido a los tiempos de retraso entre spindles que aparecen en distintos sensores.

Éste último es un problema que hay que resolver para una aplicación eficaz del ICA en la detección de spindles.

3.4 CONCLUSIONES

En este capítulo hemos visto que el ICA puede convertirse pronto en una herramienta muy útil para el preprocesado de señales biológicas registradas mediante varios (o muchos) sensores simultáneamente.

En numerosas medidas médicas se cumplen las condiciones exigidas por el ICA para su aplicación a las mismas. Esto permite utilizar el ICA para separar las señales que llegan a los sensores en componentes independientes.

Hemos visto que la aplicación del ICA tiene una doble misión. Por un lado la eliminación (mediante la deflación) de señales indeseadas (artefactos) que impiden una visión clara de las señales de interés. Por otra parte el ICA es útil para el análisis de las propias señales de interés, permitiendo el estudio de diversos fenómenos que se producen simultáneamente en el cuerpo humano. Esto permite el estudio tanto del funcionamiento de diversos órganos como de enfermedades, lesiones o anomalías.

Hemos podido ver que el análisis ICA se aplica a los datos obtenidos a partir de medidas tales como el ECG, el EEG, el MEG o la fMRI con resultados satisfactorios en numerosos casos.

Hemos visto aplicaciones importantes, como la extracción del ECG del feto a partir de las medidas recogidas por electrodos en el vientre de la madre, el estudio del flujo cardiaco autónomo a partir del ECG, o la separación de ERPs en los EEG y MEG, además de la extracción y eliminación de artefactos en todas estas medidas.

Sin embargo, también hemos visto que no siempre son asumibles las condiciones del análisis ICA como hipótesis aplicables a los datos proporcionados por las distintas técnicas y en diversas situaciones, de manera que, a veces, es necesario un conocimiento previo de la estadística o de ciertas características de las fuentes para poder separarlas.

También hemos visto que hay aún numerosos caso en que la aplicación del ICA no proporciona resultados totalmente satisfactorios, por lo que otras técnicas deberán unirse al ICA para la identificación de las fuentes.

CAPÍTULO 4

ALGORITMOS UTILIZADOS

En este capítulo se describen los algoritmos utilizados en las simulaciones y cuyos resultados van a ser comparados en el capítulo dedicado a las mismas.

4.1 ALGORITMO DE EXTRACCIÓN EN PRESENCIA DE RUIDO GAUSSIANO

Este algoritmo es el presentado por Cruces, Cichocki y Castedo en [Cruces2000]. Se trata de un algoritmo que extrae E fuentes de una vez, siendo $E \leq N$, donde N es el número total de fuentes existentes. Combinado con la deflación sirve como algoritmo de separación.

Consideremos el caso general en que existen N fuentes y M sensores, siendo $N \leq M$. Entonces

$$\mathbf{s} = \begin{bmatrix} s_1[n] \\ \vdots \\ s_N[n] \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1[n] \\ \vdots \\ x_M[n] \end{bmatrix} \quad \mathbf{e} = \begin{bmatrix} e_1[n] \\ \vdots \\ e_M[n] \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1[n] \\ \vdots \\ y_E[n] \end{bmatrix}$$

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{e}$$

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

\mathbf{A} es de orden $M \times N$

\mathbf{W} es de orden $E \times M$

La matriz de autocorrelación de las observaciones $\mathbf{R}_{\mathbf{xx}}$ tendrá, en general, rango P , siendo $N \leq P \leq M$. Entonces, la matriz de preblanqueo \mathbf{B} es de rango $P \leq M$, y las observaciones preblanqueadas son

$$\mathbf{x}' = \mathbf{B}\mathbf{x}$$

$$\mathbf{x}' = \begin{bmatrix} x'_1 [n] \\ \vdots \\ x'_p [n] \end{bmatrix}$$

En este caso $\mathbf{R}_{\mathbf{x}\mathbf{x}'} = \mathbf{I}_p$. Además $\mathbf{y} = \mathbf{Q}\mathbf{x}' = \mathbf{Q}\mathbf{B}\mathbf{x}$, siendo \mathbf{Q} una matriz de columnas ortogonales de orden $E \leq P$. Por lo tanto, $\mathbf{W} = \mathbf{Q}\mathbf{B}$.

Es posible dividir el vector de fuentes en dos partes, de modo que

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_E \\ \mathbf{s}_R \end{bmatrix}$$

En esta notación \mathbf{s}_E es el vector de las E fuentes que pretendemos recuperar, mientras que \mathbf{s}_R es el de las restantes $N-E$ fuentes.

Consecuentemente \mathbf{A} se descompone de la forma $\mathbf{A} = [\mathbf{A}_E | \mathbf{A}_R]$.

El algoritmo asume (sin pérdida de generalidad, puesto que el escalado no influye en la recuperación de las fuentes) que las fuentes se encuentran escaladas de manera que

$$\mathbf{C}_{s,s}^{1,3} = \mathbf{S}_s$$

\mathbf{S}_s es la matriz diagonal de los signos de los cumulantes

$$[\mathbf{S}_s]_{ii} = \text{sign}([\mathbf{C}_{s,s}^{1,3}]_{ii})$$

4.1.1 Algoritmo CII de partida

El algoritmo de BSE presentado en [Cruces2000] está basado en un algoritmo de BSS previo llamado CII de los mismos autores.

El algoritmo CII (donde el número de fuentes a extraer es N) se basa en el hecho de que

$$\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y$$

es un estimador robusto local de \mathbf{A} . En consecuencia el algoritmo es, en su iteración $(n+1)$ -ésima

$$\mathbf{A}^{(n+1)} = \mathbf{A}^{(n)} - \mu^{(n)} \left(\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y - \mathbf{A}^{(n)} \right)$$

De manera equivalente es posible realizar el algoritmo buscando directamente la matriz \mathbf{W} (de orden $N \times M$). Esto se lleva a cabo teniendo en cuenta que \mathbf{W} es la pseudoinversa de \mathbf{A}

$$\mathbf{W} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H = \mathbf{A}^\#$$

Esto conduce a

$$\mathbf{W}^{(n+1)} = [\mathbf{I}_N + \mu^{(n)} (\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y - \mathbf{I}_N)]^{-1} \mathbf{W}^{(n)}$$

Este algoritmo aprovecha el hecho de que las señales Gaussianas tienen cumulantes de orden superior nulos para trabajar bien bajo condiciones de ruido Gaussiano.

4.1.2 Algoritmo de BSE a partir del algoritmo CII

Podemos escribir el algoritmo CII de la siguiente manera

$$\left[\mathbf{A}_E^{(n+1)} \middle| \mathbf{A}_R^{(n+1)} \right] = \left[\mathbf{A}_E^{(n)} \middle| \mathbf{A}_R^{(n)} \right] + \mu^{(n)} \left(\left(\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y \right)_E \left[\left(\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y \right)_R \right] - \left[\mathbf{A}_E^{(n)} \middle| \mathbf{A}_R^{(n)} \right] \right)$$

Esta ecuación se puede desacoplar y, por lo tanto, podemos hallar \mathbf{A}_E sin necesidad de hallar \mathbf{A}_R

$$\mathbf{A}_E^{(n+1)} = \mathbf{A}_E^{(n)} - \mu^{(n)} \left(\left(\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y \right)_E - \mathbf{A}_E^{(n)} \right)$$

Las salidas serán

$$\mathbf{y}^{(n)} = \mathbf{W}^{(n)} \left[\mathbf{A}_E \middle| \mathbf{A}_R \begin{bmatrix} \mathbf{s}_E \\ \mathbf{s}_R \end{bmatrix} \right] + \mathbf{W}^{(n)} \mathbf{e} = \mathbf{W}^{(n)} \mathbf{A}_E \mathbf{s}_E + \mathbf{W}^{(n)} \mathbf{A}_R \mathbf{s}_R + \mathbf{W}^{(n)} \mathbf{e}$$

Luego las salidas tienen tres componentes diferentes. La primera es la debida a las E fuentes que queremos extraer. La segunda se debe a las restantes fuentes. La tercera se debe al ruido.

Puesto que se pretenden extraer las fuentes \mathbf{s}_E , $\mathbf{W}^{(n)}$ deberá cumplir los dos siguientes requisitos:

1. Cada una de las E fuentes deseadas deberá aparecer en una salida. Para ello

$$\mathbf{W}^{(n)} \mathbf{A}_E^{(n)} = \mathbf{I}_N$$

2. Las fuentes no deseadas y el ruido deberán tener una contribución mínima en las salidas. Para ello

Existen ahora dos opciones: realizar el preblanqueo de las observaciones o no realizarlo.

Si no realizamos el preblanqueo, la solución al problema se obtiene minimizando el error cuadrático medio entre las fuentes estimadas $(\mathbf{A}_E^{(n)})^\# \mathbf{A}_E \mathbf{s}_E$ y las salidas \mathbf{y} . Esto es equivalente a minimizar la potencia de las salidas sujeto a la condición (1).

$$\min_{\mathbf{W}} \left(\text{tr} \{ \mathbf{R}_{yy} \} \right) \quad \text{s.a.} \quad \mathbf{W}^{(n)} \mathbf{A}_E^{(n)} = \mathbf{I}_E$$

La solución viene dada por

$$\mathbf{W}^{(n)} = \left(\left(\mathbf{A}_E^{(n)} \right)^H \mathbf{R}_{xx}^\# \mathbf{A}_E^{(n)} \right)^{-1} \left(\mathbf{A}_E^{(n)} \right)^H \mathbf{R}_{xx}^\#$$

Por lo tanto, el algoritmo vendrá dado por las ecuaciones

$$\begin{aligned} \mathbf{A}_E^{(n+1)} &= \mathbf{A}_E^{(n)} - \mu^{(n)} \left(\left(\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y \right)_E - \mathbf{A}_E^{(n)} \right) \\ \mathbf{W}^{(n+1)} &= \left(\left(\mathbf{A}_E^{(n+1)} \right)^H \mathbf{R}_{xx}^\# \mathbf{A}_E^{(n+1)} \right)^{-1} \left(\mathbf{A}_E^{(n+1)} \right)^H \mathbf{R}_{xx}^\# \end{aligned}$$

La segunda ecuación sólo se utiliza en el último paso.

Sin embargo este algoritmo es inestable cuando algunas columnas de \mathbf{A}_E son dependientes, ya que, en ese caso, no se cumple la condición (1). Esto se refleja en la inversión que hay que realizar para el cálculo de $\mathbf{W}^{(n+1)}$, la cual diverge.

La manera de solucionar este problema es preblanquear las observaciones. Veamos:

$$\mathbf{x}' = \mathbf{B}\mathbf{x} = \mathbf{B}\mathbf{A}\mathbf{s} = \mathbf{V}\mathbf{s}$$

Gracias a la ortogonalización introducida por \mathbf{B} , \mathbf{V} es una matriz de columnas ortogonales.

Por otro lado

$$\mathbf{y} = \mathbf{Q}\mathbf{x}'$$

Por lo tanto, las ecuaciones del nuevo algoritmo (teniendo en cuenta que $\mathbf{R}_{xx} = \mathbf{I}_p$) serán:

$$\begin{aligned} \mathbf{V}_E^{(n+1)} &= \mathbf{V}_E^{(n)} + \mu^{(n)} \left(\left(\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y \right)_E - \mathbf{V}_E^{(n)} \right) \\ \mathbf{Q}_E^{(n+1)} &= \left(\mathbf{V}_E^{(n+1)} \right)^\# \end{aligned}$$

En resumen, se trata de un algoritmo cuyo paso previo es el preblanqueo de las observaciones y que extrae un número arbitrario E de fuentes.

4.2 ALGORITMO DE BSE DE FUENTES DE INTERÉS

El siguiente algoritmo que vamos a ver es el presentado por Cruces, Cichocki, Amari y Castedo en [Cruces2001].

Se trata de un algoritmo de extracción que, como el anterior, extrae E fuentes de las N existentes. La diferencia entre ambos es que el algoritmo que se va a describir en este apartado utiliza una función contraste que fuerza el hecho de que las fuentes extraídas sean fuentes de interés (es decir, artefactos o las señales objeto de estudio) y no ruido.

Anteriormente otros autores habían propuesto algoritmos que utilizaban propiedades estadísticas (conocidas a priori) de las señales de interés para forzar a recuperar éstas y no otras.

El avance del algoritmo de [Cruces2001] es que no precisa del conocimiento previo de propiedades estadísticas de las señales deseadas, sino que tiene en cuenta el hecho de que se pretenden encontrar señales no Gaussianas (ya que, en general, el ruido es Gaussiano).

La medida natural de la no gaussianidad de una variable aleatoria es su kurtosis normalizada, por lo que el algoritmo buscará la extracción de las fuentes con mayor valor absoluto de sus kurtosis normalizadas.

El primer paso del algoritmo es el preblanqueo de las observaciones. Por lo tanto, las observaciones y las salidas serán

$$\mathbf{x}' = \mathbf{BAs} = \mathbf{Vs}$$

$$\mathbf{y} = \mathbf{Qx}' = \mathbf{QVs} = \mathbf{Gs}$$

Denotamos por K_a la kurtosis no normalizada de una señal real $a[n]$ (su cumulante de cuarto orden)

$$K_a = E[a^4] - 3(E[a^2])^2$$

El algoritmo se basa en el siguiente teorema

Teorema 4.1.: Para una ordenación decreciente de las fuentes en función del valor absoluto de kurtosis

$$|K_{s_1}| \geq |K_{s_2}| \geq \dots \geq |K_{s_E}| > |K_{s_{E+1}}| \geq \dots \geq |K_{s_N}|$$

si $|K_{s_E}| \neq 0$ la siguiente función

$$\Psi(\mathbf{Y}) = \sum_{i=1}^E |K_{y_i}| \quad \text{s.a. } \mathbf{Q}\mathbf{Q}^H = \mathbf{I}_E$$

es una función contraste cuyo máximo global conduce a la extracción de las primeras E fuentes con mayores kurtosis normalizadas.

Veamos que esto es cierto. Puesto que las filas de \mathbf{Q} son ortogonales y las columnas de \mathbf{V} son también ortogonales (si $P = N$ \mathbf{V} es ortogonal y si $E = N$ \mathbf{Q} es ortogonal) las filas de \mathbf{G} son también ortogonales. Es decir, $\mathbf{G}\mathbf{G}^H = \mathbf{I}_E$.

Por otra parte

$$\begin{aligned} \Psi(\mathbf{Y}) &= \sum_{i=1}^E |K_{y_i}| = \sum_{i=1}^E \text{sign}(K_{y_i}) K_{y_i} = \sum_{i=1}^E \text{sign}(K_{y_i}) \sum_{j=1}^N G_{ij}^4 K_{s_j} \leq \\ &\leq \sum_{i=1}^E \sum_{j=1}^N G_{ij}^4 |K_{s_j}| = \sum_{j=1}^N |K_{s_j}| \sum_{i=1}^E G_{ij}^4 \leq \sum_{j=1}^N |K_{s_j}| \sum_{i=1}^E G_{ij}^2 = \text{tr}\{\mathbf{G}\mathbf{L}_K\mathbf{G}^H\} \end{aligned}$$

\mathbf{L}_K es una matriz diagonal cuyos elementos diagonales son $[\mathbf{L}_K]_{ii} = |K_{s_i}|$. La última desigualdad viene dada por el hecho de que \mathbf{G} es de filas ortogonales y $|G_{ij}| \leq 1 \forall ij$.

Según el teorema de Poincaré, y teniendo en cuenta el orden de las fuentes, el máximo de $\text{tr}\{\mathbf{G}\mathbf{L}_K\mathbf{G}^H\}$ sujeto a $\mathbf{G}\mathbf{G}^H = \mathbf{I}_E$ es

$$\sum_{j=1}^E |K_{s_j}|$$

y se consigue cuando

$$\begin{aligned} G_{ij} &= 0 \quad \forall j > E \\ \sum_{i=1}^E G_{ij}^2 &= 1 \quad \forall j \leq E \end{aligned}$$

Además, para que $\Psi(\mathbf{Y}) = \text{tr}\{\mathbf{G}\mathbf{L}_K\mathbf{G}^H\}$ es necesario que G_{ij} sea ± 1 o 0 cuando $|K_{s_j}| \neq 0$. Puesto que $|K_{s_j}| \neq 0 \forall j \leq E$ la submatriz formada por las E primeras columnas de \mathbf{G} se puede reducir, por permutaciones, a la identidad \mathbf{I}_E . Por lo tanto $\mathbf{G} = [\mathbf{I}_E | \mathbf{0}]$, con lo que se extraen las E primeras fuentes.

Para realizar la maximización de la función contraste propuesta, en [Cruces2001] se utiliza el gradiente natural. Ello conduce definitivamente al algoritmo

$$\mathbf{Q}^{(n+1)} = \mathbf{Q}^{(n)} + \mu (\mathbf{S}_y \mathbf{C}_{y,x}^{3,1} - \mathbf{C}_{y,y}^{1,3} \mathbf{S}_y \mathbf{Q}^{(n)})$$

Donde

$$\begin{aligned} |s_{y_i}|_{ii} &= |\text{sign}(K_{y_i})|_{ii} \\ \mathbf{C}_{y,x}^{3,1} &= [\mathbf{C}_{x,y}^{1,3}]^T \end{aligned}$$

Por lo tanto, éste es también un algoritmo de extracción de E fuentes con prewhitening, pero que, a diferencia del anterior, no extrae cualesquiera E fuentes, sino las E fuentes con mayores kurtosis (es decir, las menos Gaussianas).

4.3 ALGORITMO FAST-ICA

El algoritmo FAST-ICA fue propuesto por Hyvärinen y Oja y es el utilizado por Vigário en [Vigário97].

Se trata de un algoritmo de extracción de una sola fuente. Es más antiguo y limitado que los dos anteriores que hemos visto, puesto que en los casos anteriores era posible extraer un número arbitrario de E fuentes (con $E \leq N$).

Vamos a ver, además, que el algoritmo FAST-ICA es (salvo mínimas diferencias) un caso particular del algoritmo visto en el apartado 4.1. de Cruces, Cichocki y Castedo.

En el capítulo de simulaciones se utiliza una aplicación para MATLAB denominada FASTICA que realiza tanto el algoritmo FAST-ICA como otras variantes del mismo que además están generalizadas para extraer E fuentes.

También el algoritmo FAST-ICA tiene como primer paso (de la misma manera que los otros dos anteriores ya vistos) el preblanqueo o decorrelación de las observaciones.

Como $E = 1$ la matriz de extracción a calcular será \mathbf{Q}_1 (que consistirá en un vector de una fila y P columnas y de módulo unidad, ya que, debido al preblanqueo $\mathbf{Q}_1 \mathbf{Q}_1^H = \mathbf{I}_1 = 1$).

Las salidas serán

$$\mathbf{y} = \mathbf{Q}_1 \mathbf{x}'$$

De modo que

$$\begin{aligned} E[\mathbf{y}^2] &= 1 \\ K_y &= E[\mathbf{y}^4] - 3 \end{aligned}$$

El algoritmo busca la maximización de la kurtosis de la salida. El gradiente, con respecto a \mathbf{Q}_1 , de dicha kurtosis, es

$$4E[\mathbf{x}'(\mathbf{Q}_1 \mathbf{x}')^3]$$

Esto conduce a la siguiente ecuación del algoritmo

$$[\mathbf{Q}_1^{(n+1)}]^H = E[\mathbf{x}'(\mathbf{Q}_1^{(n)} \mathbf{x}')^3] - 3[\mathbf{Q}_1^{(n)}]^H$$

La esperanza es estimada utilizando el promedio temporal.

El algoritmo, para mantener el módulo unidad de \mathbf{Q}_1 , introduce otra actualización en cada paso:

$$\mathbf{Q}_1^{(n+1)} \leftarrow \frac{\mathbf{Q}_1^{(n+1)}}{\sqrt{\mathbf{Q}_1^{(n+1)}(\mathbf{Q}_1^{(n+1)})^H}}$$

La medida que utiliza este algoritmo para decidir si se está lo suficientemente cerca de la separación es

$$\left| \mathbf{Q}_1^{(n+1)}(\mathbf{Q}_1^{(n)})^H \right|$$

Si este valor es suficientemente cercano a 1 se detiene el algoritmo.

4.3.1 Relación con el algoritmo del apartado 4.1.

Vamos a comprobar en este subapartado cómo el algoritmo FAST-ICA resulta ser un caso particular del algoritmo propuesto en [Cruces2000] y que hemos estudiado en el apartado 4.1.

Aunque existen algunas diferencias poco significativas, comprobaremos que, en cada iteración, el cálculo de $\mathbf{Q}_1^{(n+1)}$ en ambos algoritmos es prácticamente el mismo cuando (en el caso del algoritmo del apartado 4.1.) $\mu = 1$, con lo que podemos decir que el algoritmo FAST-ICA es el caso particular del algoritmo propuesto en [Cruces2000] en que tanto E como μ son iguales a 1.

Por un lado, el algoritmo FAST-ICA realiza el siguiente cálculo:

$$\begin{aligned} (\mathbf{Q}_1^{(n+1)})^H &= \mathbb{E} \left[\mathbf{x}' (\mathbf{Q}_1^{(n)} \mathbf{x}')^3 \right] - 3(\mathbf{Q}_1^{(n)})^H \\ (\mathbf{Q}_1^{(n+1)})^H &\leftarrow \frac{(\mathbf{Q}_1^{(n+1)})^H}{\sqrt{\mathbf{Q}_1^{(n+1)} (\mathbf{Q}_1^{(n+1)})^H}} \end{aligned}$$

Por otro lado, el algoritmo del apartado 4.1., cuando $\mu = 1$ y $E = 1$ realiza el siguiente cálculo

$$\begin{aligned} \mathbf{V}_1^{(n+1)} &= \mathbf{C}_{x',y}^{1,3} \mathbf{S}_y \\ \mathbf{Q}_1^{(n+1)} &= \frac{(\mathbf{V}_1^{(n+1)})^H}{(\mathbf{V}_1^{(n+1)})^H \mathbf{V}_1^{(n+1)}} \\ (\mathbf{Q}_1^{(n+1)})^H &= \frac{\mathbf{V}_1^{(n+1)}}{(\mathbf{V}_1^{(n+1)})^H \mathbf{V}_1^{(n+1)}} \end{aligned}$$

Una de las diferencias está en la normalización: mientras que FAST-ICA normaliza dividiendo por el módulo, el otro algoritmo divide por el cuadrado del módulo.

Ahora veremos que el cálculo de $(\mathbf{Q}_1^{(n+1)})^H$ de FAST-ICA y el de $\mathbf{V}_1^{(n+1)}$ del algoritmo del apartado 4.1. es casi idéntico

$$\begin{aligned} \mathbf{V}_1^{(n+1)} &= \mathbf{C}_{x',y}^{1,3} \mathbf{S}_y = \left(\mathbb{E} [\mathbf{x}' \mathbf{y}^3] - 3\mathbb{E} [\mathbf{y}^2] \mathbb{E} [\mathbf{x}' \mathbf{y}] \right) \mathbf{S}_y = \\ &= \left(\mathbb{E} \left[\mathbf{x}' (\mathbf{Q}_1^{(n)} \mathbf{x}')^3 \right] - 3(\mathbf{Q}_1^{(n)})^H \right) \mathbf{S}_y \end{aligned}$$

Por lo tanto la otra diferencia existente entre ambos algoritmos es la ausencia del término \mathbf{S}_y en la ecuación del algoritmo FAST-ICA.

4.4 ALGORITMO INFOMAX

Por último veremos el algoritmo INFOMAX de Bell y Sejnowski. Se trata de un algoritmo de separación de fuentes (BSS).

Ya hemos estudiado en el capítulo 2 el criterio INFOMAX (de maximización de información), introducido en el análisis ICA por Bell y Sejnowski. Ahora veremos la solución propuesta por Amari, que conduce al algoritmo INFOMAX.

Esta solución se basa en la utilización del gradiente natural para buscar el máximo de la función contraste, al contrario que la propuesta por los propios Bell y Sejnowski, basada en el gradiente convencional.

Ambas soluciones podemos encontrarlas en [Bell2000] y en [Cruces99].

Hemos visto en el capítulo 2 que el criterio de separación INFOMAX es equivalente al criterio ENTMAX (de maximización de entropía de las salidas) debido a que

$$I(\mathbf{Y}; \mathbf{X}) = h(\mathbf{Y}) - h(\mathbf{Y}|\mathbf{X})$$

y a que $h(\mathbf{Y}|\mathbf{X})$ es independiente de \mathbf{W} .

También vimos que una maximización directa de $h(\mathbf{Y})$ no era conveniente por no estar acotada. Esto nos llevó a maximizar la entropía de \mathbf{Z} , donde se cumplía la relación

$$\mathbf{z} = \mathbf{q}(\mathbf{y})$$

siendo

$$z_i = q_i(y_i) = \int_{-\infty}^{\infty} p_{S_i}(y_i) dy_i \quad i = 1, \dots, N$$

Por lo tanto, la función a maximizar es

$$h(\mathbf{Z}) = -E[\log(p_{\mathbf{Z}}(\mathbf{z}))]$$

Pero

$$p_{\mathbf{Z}}(\mathbf{z}) = \frac{p_{\mathbf{Y}}(\mathbf{y})}{|J|}$$

donde J es el determinante de la matriz $[dz_i/dy_j]_{ij}$.

Como

$$\begin{aligned} \frac{dz_i}{dy_i} &= p_{S_i}(y_i) \\ \frac{dz_i}{dy_j} &= 0 \quad \forall j \neq i \end{aligned}$$

se tiene que

$$J = \prod_{i=1}^N p_{S_i}(y_i) = p_{\mathbf{S}}(\mathbf{y}) \geq 0$$

Luego

$$h(\mathbf{Z}) = -E \left[\log \left(\frac{p_{\mathbf{Y}}(\mathbf{y})}{p_{\mathbf{S}}(\mathbf{y})} \right) \right] = -E[\log(p_{\mathbf{Y}}(\mathbf{y}))] + E[\log(p_{\mathbf{S}}(\mathbf{y}))]$$

Además

$$p_{\mathbf{Y}}(\mathbf{y}) = \frac{p_{\mathbf{X}}(\mathbf{x})}{|\det(\mathbf{W})|}$$

Por lo que

$$h(\mathbf{Z}) = h(\mathbf{X}) + \log|\det(\mathbf{W})| + E[\log(p_{\mathbf{S}}(\mathbf{y}))]$$

Sin embargo, $h(\mathbf{X})$ tampoco depende de \mathbf{W} , según [Bell2000]. Por lo tanto, la función contraste será

$$\log|\det(\mathbf{W})| + E[\log(p_{\mathbf{S}}(\mathbf{y}))]$$

Como no conocemos $p_{\mathbf{S}}(\mathbf{y})$ tenemos que estimarla. Por lo tanto, llegamos de nuevo a la conclusión de que el criterio INFOMAX (o ENTMAX) es coincide con el de máxima verosimilitud (ML). la función contraste definitiva es, pues

$$\Psi(\mathbf{Y}) = \log|\det(\mathbf{W})| + E[\log(p_{\hat{\mathbf{S}}}(\mathbf{y}))]$$

Derivando con respecto a \mathbf{W} hallamos el gradiente convencional

$$\begin{aligned} \frac{\partial \log|\det(\mathbf{W})|}{\partial \mathbf{W}} &= (\mathbf{W}^T)^{-1} \\ \frac{\partial E[\log(p_{\hat{\mathbf{S}}}(\mathbf{y}))]}{\partial \mathbf{W}} &= -\mathbf{R}_{\hat{\mathbf{x}}} \\ \mathbf{f}(\mathbf{y}) &= \left[-\frac{dp_{\hat{\mathbf{S}}_1}(y_1)/dy_1}{p_{\hat{\mathbf{S}}_1}(y_1)}, \dots, -\frac{dp_{\hat{\mathbf{S}}_N}(y_N)/dy_N}{p_{\hat{\mathbf{S}}_N}(y_N)} \right] \\ \mathbf{R}_{\hat{\mathbf{x}}} &= E[\mathbf{f}(\mathbf{Y})\mathbf{X}^T] \end{aligned}$$

$\mathbf{R}_{\hat{\mathbf{x}}}$ es una matriz de correlación no lineal. Vamos a ver ahora cómo se llega a ella

$$\begin{aligned}
 y_i &= \sum_{k=1}^N W_{ik} x_k \\
 \frac{\partial \log [p_{\hat{s}_i}(y_i)]}{\partial W_{jk}} &= 0 \quad \forall j \neq i \\
 \frac{\partial \log [p_{\hat{s}_i}(y_i)]}{\partial W_{ik}} &= \frac{dp_{\hat{s}_i}(y_i)/dy_i}{p_{\hat{s}_i}(y_i)} \frac{\partial y_i}{\partial W_{ik}} = \frac{dp_{\hat{s}_i}(y_i)/dy_i}{p_{\hat{s}_i}(y_i)} x_k \\
 \frac{\partial E[\log(p_{\hat{s}}(\mathbf{y}))]}{\partial \mathbf{W}} &= E \left[\frac{\partial \log(p_{\hat{s}}(\mathbf{y}))}{\partial \mathbf{W}} \right] = E \left[\left[\frac{\partial \log [p_{\hat{s}_i}(y_i)]}{\partial W_{ik}} \right]_{ik} \right] = \\
 &= E \left[\left[\frac{dp_{\hat{s}_i}(y_i)/dy_i}{p_{\hat{s}_i}(y_i)} x_k \right]_{ik} \right] = E[-\mathbf{f}(\mathbf{Y})\mathbf{X}^T] = -\mathbf{R}_{fx}
 \end{aligned}$$

Por lo tanto, el gradiente convencional será

$$\frac{\partial \Psi(\mathbf{Y})}{\partial \mathbf{W}} = (\mathbf{W}^T)^{-1} - \mathbf{R}_{fx} = (\mathbf{W}^T)^{-1} - E[\mathbf{f}(\mathbf{Y})\mathbf{X}^T]$$

Para obtener el gradiente natural se multiplica por $\mathbf{W}^T\mathbf{W}$, según se puede ver en [Bell2000]

$$\begin{aligned}
 \nabla \Psi(\mathbf{Y}) &= (\mathbf{W}^T)^{-1} \mathbf{W}^T \mathbf{W} - \mathbf{R}_{fx} \mathbf{W}^T \mathbf{W} = \mathbf{W} - E[\mathbf{f}(\mathbf{Y})\mathbf{X}^T \mathbf{W}^T] \mathbf{W} = \\
 &= \mathbf{W} - E[\mathbf{f}(\mathbf{Y})\mathbf{Y}^T] \mathbf{W} = (\mathbf{I} - \mathbf{R}_{fy}) \mathbf{W}
 \end{aligned}$$

Por lo tanto, el algoritmo asociado será

$$\mathbf{W}^{(n+1)} = \mathbf{W}^{(n)} - \mu(\mathbf{R}_{fy} - \mathbf{I})\mathbf{W}^{(n)}$$

Utilizando el estimador instantáneo de \mathbf{R}_{fy}

$$\mathbf{R}_{fy} \approx \mathbf{f}(\mathbf{y})\mathbf{y}^T$$

tenemos el algoritmo estocástico

$$\mathbf{W}^{(n+1)} = \mathbf{W}^{(n)} - \mu(\mathbf{f}(\mathbf{y})\mathbf{y}^T - \mathbf{I})\mathbf{W}^{(n)}$$

En realidad el algoritmo suele utilizar también el prewhitening de las observaciones, por lo que lo visto para \mathbf{W} normalmente se realiza con \mathbf{Q} .

El problema es que no conocemos las $p_{s_i}(y_i)$. Ni siquiera tenemos una idea para poder estimarlas. Por lo tanto, para poder utilizar el algoritmo es necesario encontrar una $\mathbf{q}(\mathbf{y})$ adecuada.

Según [Bell2000] hay varias opciones. Una de ellas es la utilizada por Bell y Sejnowski, consistente en tomar $q(y_i) = q(y_i)$, siendo $q(y_i)$ una función de distribución cuya derivada es una f.d.p. superGaussiana (de kurtosis > 0). Esto conduce a la separación de las fuentes superGaussianas. De igual modo, si $q(y_i)$ es subGaussiana (es decir, de kurtosis < 0) se produce la separación de las fuentes subGaussianas.

Otra opción (propuesta por Amari) consiste en utilizar la función $q(y_i) = (1 + e^{-y_i})^{-1}$, que separa las fuentes superGaussianas. Realmente éste es un caso particular del propuesto por Bell y Sejnowski.

Una tercera opción es la propuesta por Girolami, consistente en calcular la kurtosis de las salidas y_i y, consecuentemente, utilizar funciones $q_i(y_i)$ que sean superGaussianas o subGaussianas. Es decir, en cada iteración las $q_i(y_i)$ serán unas iguales a una función superGaussiana y otras iguales a una función subGaussiana. Además, entre una iteración y la siguiente habrá cambios.

La regla utilizada es la siguiente

$$f_i(y_i) = \begin{cases} -\tanh(y_i) - y_i & \text{si kurtosis}(y_i) > 0 \\ \tanh(y_i) - y_i & \text{si kurtosis}(y_i) < 0 \end{cases}$$

Este método es el conocido como ICA extendido o INFOMAX extendido. Tanto el uso de $(1 + e^{-y_i})^{-1}$ como el de ICA extendido son posibles en el toolbox para MATLAB de Scott Makeig et al. utilizado en el capítulo de simulaciones, siendo el primer método el utilizado por defecto.

4.5 CONCLUSIONES

En este capítulo hemos visto cuatro algoritmos que van a ser utilizados posteriormente en las simulaciones y que serán comparados. Tres de ellos (los dos basados en los cumulantes y el FAST-ICA) son de extracción (BSE), mientras que el cuarto (INFOMAX) es de separación (BSS).

El primer algoritmo de extracción que hemos visto está basado en un algoritmo previo de separación (CII) que utiliza un cumulante cruzado de cuarto orden para estimar la matriz de mezclas. A partir de ahí, comprobando que las columnas de la matriz de mezclas están desacopladas en la ecuación del algoritmo de separación, vimos

que podemos hallar exclusivamente un número E de columnas (y, por lo tanto, extraer un número E de fuentes).

También hemos visto que el preblanqueo de las observaciones es una condición necesaria para poder llevar a cabo los tres algoritmos de extracción estudiados.

El segundo algoritmo estudiado se basa en el hecho de que la suma de los valores absolutos de las kurtosis de las salidas es una función contraste. Este algoritmo tiene la ventaja de que extrae las fuentes con mayores kurtosis, que son las menos Gaussianas.

El algoritmo FAST-ICA es un algoritmo de extracción que sólo extrae una fuente. Hemos visto que se trata de un caso particular del primer algoritmo (aunque con ligeras y poco significativas diferencias).

El último algoritmo estudiado (el INFOMAX) es un algoritmo de separación consistente en la utilización del gradiente natural para la maximización de la función contraste surgida del criterio INFOMAX, anteriormente estudiado en el capítulo 2. Aunque se trata de un criterio que necesita la estimación de las f.d.p. de las fuentes, el algoritmo sólo necesita de la utilización de una función de distribución cuya derivada sea una f.d.p. superGaussiana o subGaussiana para separar las fuentes superGaussianas o subGaussianas, respectivamente.

CAPÍTULO 5

SOFTWARE EMPLEADO

En este capítulo se realiza una descripción de los programas de ordenador utilizados en las simulaciones. Estos programas se agrupan en cuatro bloques, cada uno de los cuales se centra en uno de los algoritmos de separación y extracción que hemos visto en el capítulo anterior. Se trata de programas para MATLAB. Es decir, archivos *.m.

Los dos bloques más sencillos son los centrados en los dos primeros algoritmos que hemos visto (los algoritmos de [Cruces2000] y de [Cruces2001]), ya que cada uno de ellos sólo contiene dos programas: uno contiene el algoritmo de extracción y el otro la deflación y la construcción de las salidas.

El tercer bloque es el paquete denominado **FASTICA** y que está basado, obviamente, en el algoritmo FAST-ICA. Este toolbox tiene la ventaja de poseer un interfaz gráfico, con lo que el usuario se ahorra cualquier tipo de programación o llamadas a funciones.

Por último se encuentra el toolbox **MATLAB ICA/EEG**. Éste es el toolbox en el que está basado el proyecto, debido a que posee numerosas funciones de representación gráfica de los datos y de las componentes independientes ya separadas, así como de sus proyecciones sobre las observaciones. Para una utilización más cómoda de las funciones de que dispone este toolbox, en este proyecto se han desarrollado pequeños programas que realizan llamadas a dichas funciones.

5.1 EXTRACT_QN

El programa (archivo .m de MATLAB) que lleva a cabo el algoritmo descrito en el apartado 4.1. (el algoritmo de [Cruces2000]) se denomina **extract_QN**. Dispone de un programa auxiliar denominado **bsseQN** que realiza la extracción llamando a **extract_QN** y la deflación, para así extraer cuantas componentes solicite el usuario. Los autores son Cruces, Cichocki y Castedo.

5.1.1 bsseQN

Este programa requiere de la existencia de una variable llamada "x" en el espacio de trabajo de MATLAB que contenga en cada fila cada una de las observaciones muestreadas en el tiempo.

El programa solicita al usuario el valor de stop (valor del grado de convergencia para el que se detiene el algoritmo de extracción). También solicita al usuario el número de componentes principales a que quiere reducir las observaciones, el número total de componentes a extraer y el número de componentes a extraer en cada paso.

Posteriormente realiza una primera llamada a **extract_QN** para que lleve a cabo el preblanqueo y la primera extracción. Después realiza la deflación e inicia un bucle para extraer el resto de componentes (ya sin realizar preblanqueo).

Por último guarda la matriz de prewhitening en la variable "sphere" y la de separación (la que hemos llamado **Q** en los capítulos anteriores) en "weights". Estas dos variables, junto con la de los datos, serán almacenadas en un archivo para una posterior obtención y visualización de las componentes independientes en el toolbox MATLAB ICA/EEG. No se almacenan las componentes independientes debido a la gran cantidad de espacio en disco que ocupan tanto los datos como las componentes.

5.1.2 extract_QN

Éste es el programa que lleva a cabo la extracción de las E componentes de cada paso.

Como argumentos de entrada tiene:

- las observaciones (o bien las observaciones después de haber sufrido una o varias deflaciones)
- el número de componentes a extraer (es decir E)
- el valor que detiene el algoritmo según el grado de convergencia
- un parámetro que indica si se quiere hacer prewhitening o no, y, en caso de hacerlo, a cuantas componentes principales se quieren reducir las observaciones.

Este último parámetro tiene una ligera modificación con respecto al original, ya que el original no ofrecía la posibilidad de reducir.

Las salidas son:

- la matriz de extracción una vez preblanqueados los datos (es decir, \mathbf{Q}_E)
- las columnas estimadas de la matriz de mezclas incluyendo el prewhitening (es decir, \mathbf{V}_E)
- las salidas (las componentes independientes; es decir, \mathbf{y})
- las entradas preblanqueadas (es decir, \mathbf{x}').

Este algoritmo puede ser modificado de manera que el estimador de la matriz de mezclas no sea

$$\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y$$

sino una combinación lineal (sumando 1 los coeficientes) de

$$\mathbf{C}_{x,y}^{1,1} \mathbf{S}_y^1$$

$$\mathbf{C}_{x,y}^{1,2} \mathbf{S}_y^2$$

$$\mathbf{C}_{x,y}^{1,3} \mathbf{S}_y^3$$

siendo

$$\mathbf{S}_y^k$$

diagonal y cumpliendo

$$\left[\mathbf{S}_y^k \right]_{ii} = \text{sign} \left(\left[\mathbf{C}_{x,y}^{1,k} \right]_{ii} \right)$$

5.2 EXTRACT2

Se denomina **extract2** el programa que lleva a cabo el algoritmo descrito en el apartado 4.2. (el algoritmo de [Cruces2001]). Los autores de este programa son Cruces y Cichocki.

Al igual que `extract_QN`, dispone de un programa auxiliar que realiza la deflación y llama a `extract2` para extraer todas las componentes solicitadas por el usuario. Ese programa auxiliar se denomina **bsse2**.

Tanto `bsse2` como `extract2` son prácticamente iguales a `bsseQN` y `extract_QN`, respectivamente, con la diferencia del algoritmo utilizado.

También `extract2` se puede modificar para utilizar en la estimación los cumulantes de segundo y tercer orden además del de cuarto orden.

5.3 FASTICA

El programa **fastica** lleva a cabo el algoritmo FAST-ICA y la deflación. Está elaborado por Gävert, Hurri, Särelä y Hyvärinen. La versión utilizada en este proyecto es la 2.0 beta 3.

Se trata de un programa más completo, que no sólo realiza el algoritmo de extracción FAST-ICA descrito en el capítulo anterior, sino que puede realizar variaciones del mismo (utilizando distintas funciones para la actualización de la matriz de extracción) y puede llevar a cabo, directamente, la separación (BSS y no BSE con deflación) de las N fuentes.

Dispone además de una interfaz gráfica (implementada mediante el programa **fasticag**) que facilita la labor al usuario, de modo que, una vez cargados los datos en una variable del espacio de trabajo de MATLAB, sólo hay que ejecutar `fasticag`, apareciendo una ventana desde la cual se definen los datos y se realiza el prewhitening y la extracción o separación de las fuentes.

Una vez cargados los datos, desde la ventana de `fasticag` se permite reducir su dimensión (Dimension control) al hacer prewhitening. También es posible trasponer los datos (en caso de que la variable que hayamos elegido tenga las observaciones por columnas y no por filas).

En la figura 5-1 se muestra la ventana que ofrece fasticag para la realización del análisis ICA. Al pulsar el botón de carga de los datos aparece un cuadro de diálogo en el que se pide la variable que contiene los datos. Esta variable debe haber sido creada anteriormente y encontrarse en el espacio de trabajo de MATLAB. Como se verá en el apartado 5.4. esta variable es "x".



Figura 5-1 Ventana de fasticag

Cuando se pulsa "Reduced dim." Se calculan los autovalores y se solicita al usuario el conjunto de autovalores (desde el primero hasta el número deseado, ordenados de mayor a menor) que quiere utilizar para realizar el preblanqueo.

El cuadro de diálogo opciones avanzadas se muestra en la figura 5-2.

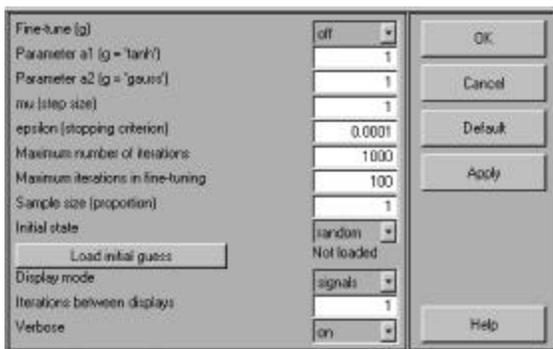


Figura 5-2 Ventana de opciones avanzadas de fasticag

Para realizar el análisis ICA existen varias opciones:

- approach: permite elegir entre hacer BSS o extraer las fuentes de 1 en 1
- number of ICs: de esta manera (tanto si hacemos BSS como si hacemos BSE con deflación) se decide cuántas fuentes se quiere extraer
- nonlinearity (g): permite elegir entre 4 algoritmos (por defecto se elige el FAST-ICA que hemos visto en el capítulo anterior)
- stabilization: es una opción que puede estar en 'on' o en 'off', y que sirve para impedir que el algoritmo no converga (para estabilizarlo).

Las opciones avanzadas son las siguientes:

- fine-tune (g): si se coloca en 'on', tras una primera convergencia, el algoritmo cambia de no linealidad, utiliza la versión estabilizada y disminuye el tamaño de paso
- parameter a1 (g = 'tanh'): esta opción permite al usuario definir el parámetro a1 en caso de que la función no lineal elegida para llevar a cabo el algoritmo sea 'tanh'
- parameter a2 (g = 'gauss'): define el parámetro a2 de la función 'gauss' en caso de que sea la función elegida
- mu (step size): es el valor de paso cuando se utiliza la versión estabilizada del algoritmo (si vale 1 no se utiliza esta versión a menos que el usuario así lo indique)
- epsilon (stopping criterion): define el valor que ha de tomar el criterio del grado de convergencia del algoritmo para detenerlo
- maximum number of iterations: se define un número máximo de iteraciones tras el cual se detienen el algoritmo
- maximum iterations in fine-tuning: máximo número de iteraciones en el ajuste fino
- sample size (proportion): permite definir el número de muestras (una proporción de los datos) que serán utilizadas en los cálculos en cada paso; se indica sólo esta proporción, pero la porción utilizada es aleatoria en cada paso

- `initial state`: permite elegir entre una matriz de extracción (o separación) inicial aleatoria o una definida por el usuario (en este último caso la matriz deberá ser cargada desde una variable que la contenga en el espacio de trabajo de MATLAB)
- `display mode`: permite elegir los resultados que queremos ver durante la ejecución del algoritmo (señales extraídas representadas gráficamente, etc.)
- `iterations between displays`: mediante este parámetro el usuario decide cada cuántas iteraciones se muestran aquellos resultados que ha solicitado visualizar
- `verbose`: habilita o inhabilita los distintos mensajes que pueden aparecer durante la ejecución del algoritmo.

A diferencia de los demás programas que van a ser utilizados, `fastica` devuelve la matriz de separación para los datos originales (mientras que los otros tres programas la devuelven para los datos originales menos sus medias). Puesto que sólo se almacenan (por razones de espacio en disco) las observaciones y las matrices de separación y preblanqueo, para que el toolbox MATLAB ICA/EEG calcule correctamente las componentes, cuando se realice la extracción o separación mediante `fastica` es conveniente cargar como datos las observaciones menos sus medias. De otro modo, tras obtener las fuentes utilizando las funciones del toolbox MATLAB ICA/EEG habría que sumarles el producto de las matrices de separación y preblanqueo por la matriz de medias de las observaciones. Es decir, si $\mathbf{y} = \mathbf{Q}\mathbf{B}\mathbf{x} = \mathbf{W}\mathbf{x}$ (siendo \mathbf{x} las observaciones originales) son las fuentes separadas por `fastica`, el toolbox obtendrá $\mathbf{y}_m = \mathbf{W}\mathbf{x}_m$ (siendo \mathbf{x}_m las observaciones menos sus medias), que resultan ser las fuentes menos sus medias. Para obtener \mathbf{y} habría que sumar a \mathbf{y}_m el vector $\mathbf{W}\mathbf{m}_x$ (siendo \mathbf{m}_x el vector de medias de \mathbf{x}).

Veamos ahora los 4 algoritmos que utiliza `fastica` (sin contar sus versiones estabilizadas ni sus versiones BSS) y que se eligen mediante el valor del parámetro `nonlinearity (g)`.

5.3.1 `pow3`

Cuando el parámetro `nonlinearity (g)` se define como `'pow3'` se está eligiendola versión que ya conocemos de FAST-ICA

$$\left[\mathbf{Q}_1^{(n+1)} \right]^H = E \left[\mathbf{x}' \left(\mathbf{Q}_1^{(n)} \mathbf{x}' \right)^3 \right] - 3 \left[\mathbf{Q}_1^{(n)} \right]^H$$

Se denomina 'pow3' porque utiliza la correlación entre las entradas preblanqueadas y la salida elevada al cubo.

5.3.2 tanh

El valor 'tanh' indica que el algoritmo elegido es el siguiente

$$\left[\mathbf{Q}_1^{(n+1)} \right]^H = E \left[\mathbf{x}' \tanh \left(a_1 \mathbf{Q}_1^{(n)} \mathbf{x}' \right) \right] - a_1 E \left[1 - \left(\tanh \left(a_1 \mathbf{Q}_1^{(n)} \mathbf{x}' \right) \right)^2 \right] \left[\mathbf{Q}_1^{(n)} \right]^H$$

Obviamente su nombre viene de la utilización de la tangente hiperbólica de la salida (escalada) en lugar de su cubo.

El valor a_1 es el parámetro a1 de las opciones avanzadas disponibles en fasticag.

5.3.3 gauss

El algoritmo utilizado cuando se escoge el valor 'gauss' es el siguiente

$$\left[\mathbf{Q}_1^{(n+1)} \right]^H = E \left[\mathbf{x}' \left(\mathbf{Q}_1^{(n)} \mathbf{x}' \right) e^{-a_2 \frac{\left(\mathbf{Q}_1^{(n)} \mathbf{x}' \right)^2}{2}} \right] - E \left[\left(1 - a_2 \left(\mathbf{Q}_1^{(n)} \mathbf{x}' \right)^2 \right) e^{-a_2 \frac{\left(\mathbf{Q}_1^{(n)} \mathbf{x}' \right)^2}{2}} \right] \left[\mathbf{Q}_1^{(n)} \right]^H$$

De la utilización de la Gaussiana de la salida (que multiplica a la propia salida) viene el nombre de 'gauss' para esta opción.

5.3.4 skew

Cuando se elige la opción 'skew' se está eligiendo el siguiente algoritmo

$$\left[\mathbf{Q}_1^{(n+1)} \right]^H = E \left[\mathbf{x}' \left(\mathbf{Q}_1^{(n)} \mathbf{x}' \right)^2 \right]$$

5.4 MATLAB ICA/EEG

MATLAB ICA/EEG es un toolbox para MATLAB desarrollado por Makeig, Humphries, Enghoff, y Jung que permite representar datos procedentes de EEG o MEG

y realizar sobre ellos análisis ICA, siendo muy útil también para la representación de las componentes y sus proyecciones sobre las observaciones.

Posee una gran cantidad de funciones con una importante variedad de prestaciones. La versión utilizada en este proyecto es la 3.5.1. y fue tomada de [1].

En este proyecto se han utilizado algunas de esas funciones, tanto para la realización del análisis ICA como para la visualización de los resultados obtenidos con los otros tres algoritmos empleados.

El análisis ICA se lleva acabo mediante la función `runica`, que implementa el algoritmo INFOMAX utilizando el gradiente natural, tal como hemos visto en el capítulo anterior. Realiza prewhitening (PCA) permitiendo la reducción de la dimensión de las observaciones, y tiene como opción al algoritmo INFOMAX extendido.

Para facilitar la utilización de las funciones del toolbox que serán necesarias para la ejecución del algoritmo y la visualización de los datos y los resultados, se han desarrollado en este proyecto pequeños programas que permiten al usuario trabajar sin tener que conocer las cabeceras de dichas funciones.

El toolbox se encuentra en un directorio denominado **ica5** y los pequeños programas de utilización del toolbox se encuentran en otro directorio denominado **ica5uso**.

5.4.1 `runica`

En este subapartado vamos a ver una descripción de los parámetros de entrada y las variables de salida de la función `runica`, que implementa el algoritmo de análisis ICA.

En primer lugar veremos los parámetros de entrada. El primer parámetro (y el único obligatorio) es **data**, que es la variable que contiene los datos, siendo cada canal (cada observación o sensor) una fila. Tras `data` hay una serie de parámetros opcionales, de los cuales hay que escribir, separados por comas, el nombre del parámetro y el valor que se le da. Los más importantes son los siguientes:

- `'ncomps'`: es el número de componentes que se pretenden recuperar; por defecto es el número de observaciones

- 'pca': es el número de componentes principales que se quieren conservar; así se indica la reducción de la dimensión; por defecto, la dimensión no se reduce
- 'sphering': es un flag ('on' / 'off') que indica si se quiere realizar prewhitening; por defecto se realiza
- 'weights': es la matriz inicial de separación tras el prewhitening (es decir, \mathbf{Q}); por defecto es la matriz identidad, a menos que no se realice prewhitening, en cuyo caso se toma como matriz inicial de separación la matriz de prewhitening
- 'stop': indica el valor que ha de tomar el criterio del grado de convergencia para que se detenga el algoritmo; por defecto es 10^{-6}
- 'extended': si es distinto de 0 significa la utilización del algoritmo extendido; cuando es mayor que 0 se estima el signo cada 'extended' bloques; si es menor que 0 se supone ese número (en valor absoluto) de componentes de kurtosis negativa (subGaussianas); por defecto toma el valor 0
- 'verbose': si está en 'on' habilita los mensajes (es la opción por defecto).

Las variables de salida son las siguientes:

- weights: es la matriz de separación después del prewhitening
- sphere: es la matriz de prewhitening
- activations: es la variable que contiene las componentes independientes (las salidas)
- bias
- signs: es un vector con los signos de las componentes en ICA extendido (-1 es subGaussiana y 1 es superGaussiana)
- lrates: son los valores finales de μ correspondientes a cada una de las componentes.

5.4.2 Pequeños programas para la utilización del toolbox

Aquí se describen los pequeños programas realizados para una utilización más cómoda de las funciones del toolbox.

dadqui1

Éste es programa de adquisición de los datos.

Lo primero que hace es pedir el nombre del fichero de datos en que se encuentran las observaciones (también es posible que se encuentren en él los resultados) y cargar dicho fichero.

Después solicita el nombre de la variable que contiene los datos y define el número de sensores (canales), que es el número de filas, y el de muestras, que es el de columnas.

Posteriormente se solicita al usuario el número de intervalos en que quiere dividir los datos. Esto será utilizado para representar los datos de manera que se puedan observar cómodamente, evitando una excesiva compresión de las gráficas.

Después se solicita la frecuencia de muestreo, también útil para la representación de los datos y salidas (para etiquetar los ejes) y para el posible filtrado que se realice a los datos.

Por último el programa ofrece un filtrado opcional de los datos. Se solicita al usuario la banda de paso y se realiza el filtrado utilizando la función **eegfilt** del toolbox.

Los datos quedan almacenados en la variable "data", que será empleada por los demás programas de utilización del toolbox.

El programa de adquisición de datos define además otras variables que no deben ser borradas porque tendrán que ser utilizadas por el resto de programas.

daploti2

Este programa realiza una representación gráfica de los datos, de manera que los agrupa en gráfica de 8 canales cada una (para evitar una gráficas demasiado comprimidas verticalmente) y en cada canal representa los intervalos definidos en dadqui1 superpuestos y cada uno de un color. Se permite al usuario el empleo de zoom con el ratón.

Para la realización de las gráficas se utiliza la función **plotdat2**, que consiste en una ligera modificación de **plotdata**, una de las funciones del toolbox.

La variación de **plotdata** a **plotdat2** consiste en que, mientras la primera representa los datos en 2 columnas la segunda lo hace en una sola columna. Esto se ha

hecho así porque los datos que se emplearán posteriormente en las simulaciones son muy extensos.

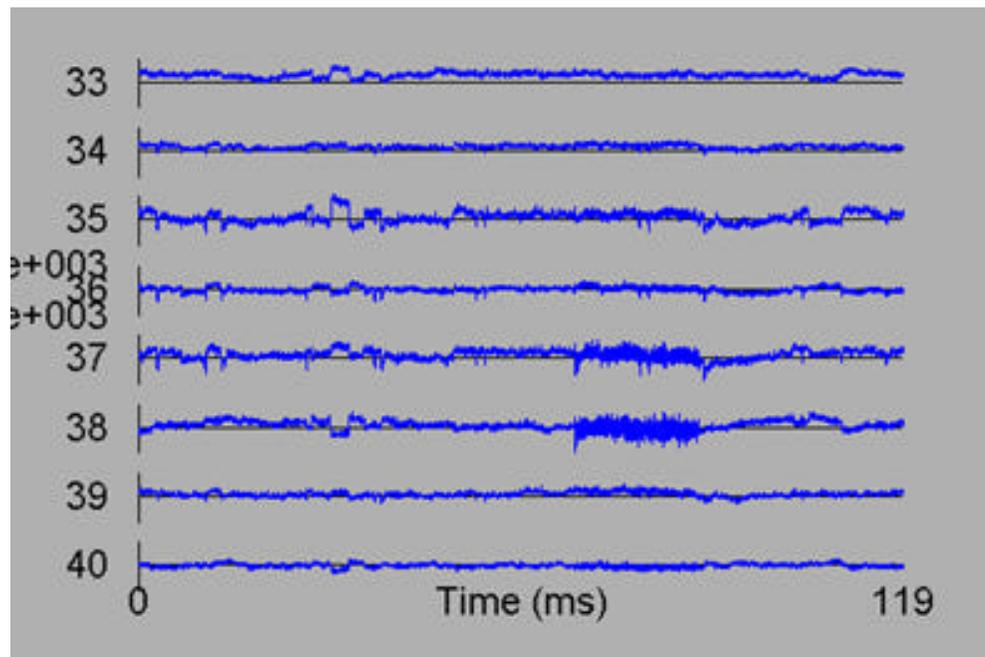


Figura 5-3 Representación gráfica realizada por dploti2

En la figura 5-3 se muestra una de las gráficas realizadas por dploti2. En este caso se tomó un solo intervalo de tiempo para todas las muestras.

datopi3

Este programa representa una superposición de los datos (de todos los sensores) en una sola gráfica y dibuja 4 mapas cerebrales para 4 instantes de tiempo que son solicitados al usuario.

Los mapas cerebrales son dibujos bidimensionales en los que se representa una vista cenital del cerebro, y con una franja de colores se indica (mediante interpolación de los datos de los sensores) el valor que toma la magnitud observada (la magnitud de las observaciones) en cada punto del dibujo. La franja de colores va del azul (el valor más negativo) al rojo (el más positivo), pasando por el amarillo.

Estas representaciones se realizan con la función **tintop2**, que es una variación que se ha realizado sobre la función **tintopo** del toolbox. La modificación consiste, únicamente, en no etiquetar el eje vertical de la gráfica como 'Potential (uV)', ya que se pretende usar el toolbox tanto para EEG como para MEG.

En la figura 5-4 se muestra un ejemplo del uso de `datopi3`.

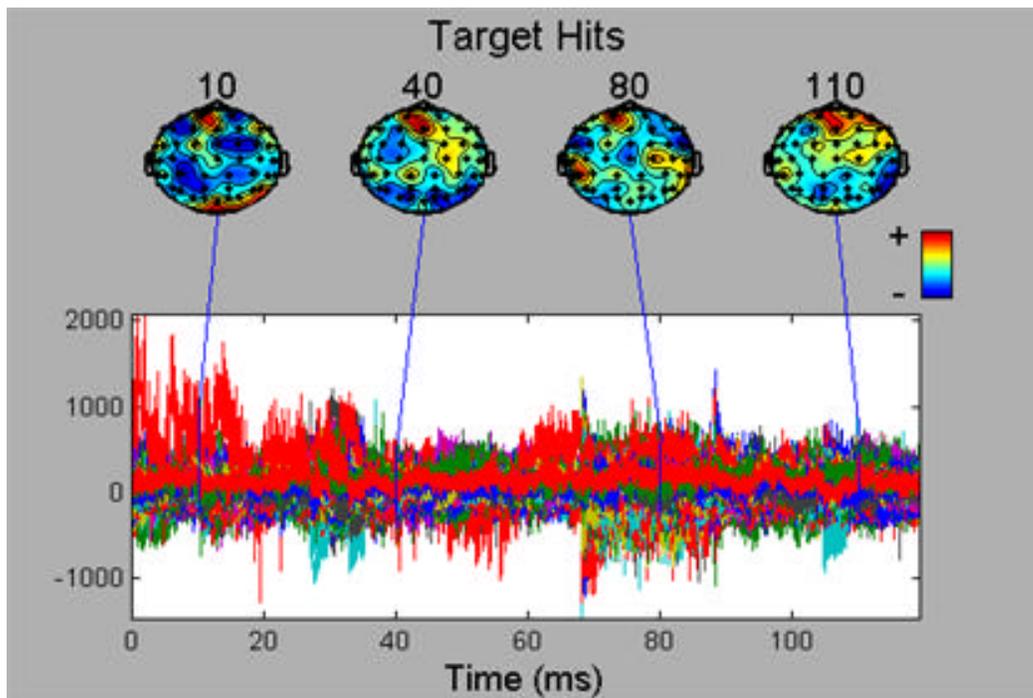


Figura 5-4 Representación generada por `datopi3`

dateegi4

Este programa realiza una representación gráfica de los datos en forma de EEG. De igual modo que `daploti2`, divide los datos en grupos de 8 canales por razones de visibilidad.

Al usuario le será solicitado un valor denominado tamaño de ventana. Esto se debe a que (de nuevo por razones de visibilidad), en pantalla solamente se representa un intervalo de los datos. El tamaño de ese intervalo (en segundos) es el que se solicita al usuario.

En las figuras que se generan para la representación de los datos se proporciona al usuario un conjunto de controles en forma de botones, de manera que el usuario puede:

- desplazar la gráfica un segundo hacia delante o hacia atrás
- desplazar la gráfica hacia delante o hacia atrás un número de segundos igual al tamaño de ventana
- desplazar la gráfica hasta un instante arbitrario

- aumentar o disminuir la escala vertical de las gráficas.

Para la realización de las representaciones gráficas `dateegi4` utiliza la función **`eegplot2`**, que es una pequeña modificación de la función **`eegplot`** que proporciona el toolbox.

La modificación consiste, exclusivamente, en eliminar la etiqueta 'uV', en numerar los canales y en indicar el tiempo en el eje temporal solamente cada 5 segundos.

También se ofrece al usuario el zoom con la utilización del ratón.

Este programa (`dateegi4`) es más rápido que `daploti2`, y es también muy cómodo para la observación de los datos.

runicai5

El programa `runicai5` se ha realizado para llevar a cabo el análisis ICA, ya sea mediante la función `runica` del toolbox, ya sea mediante cualquiera de los otros 3 algoritmos (`extract_QN`, `extract2`, `fastica`).

Por lo tanto se puede realizar el análisis ICA (por medio de alguno de los 4 algoritmos ya vistos) como continuación de la carga de los datos llevada a cabo con `dadqui1`.

Lo primero que hace `runicai5` es solicitar al usuario el algoritmo a utilizar. Si se elige `runica`, se solicita el valor de stop. Posteriormente se pregunta al usuario si quiere utilizar el algoritmo INFOMAX o el extendido. Para cualquiera de los dos casos se solicita el número de componentes principales a que se quieren reducir las observaciones.

Si se elige la opción de no reducir las observaciones se ejecuta el algoritmo elegido (INFOMAX o extendido) preguntando previamente cuántas componentes se quieren separar.

Si se elige reducir el número de componentes principales se ejecuta el algoritmo elegido utilizando el parámetro 'pca' (en vez de 'ncomps') y dándole el valor que solicitó el usuario.

Respecto a las opciones 'sphering' y 'pca' de `runica` hay que aclarar las diferencias existentes.

Hasta ahora hemos denominado sphering al prewhitening, preblanqueo o decorrelación de las observaciones y lo hemos identificado con el análisis PCA. Realmente PCA es una manera de realizar el prewhitening.

En la función runica se denomina 'sphering' a otra manera de realizar el prewhitening, que no coincide con el análisis PCA. Vamos a ver ahora en qué consiste este otro método de preblanqueo.

Supongamos que tenemos una matriz \mathbf{M} cuadrada invertible. Si su inversa es \mathbf{M}^{-1} tenemos que

$$\mathbf{M}^{-1} = \mathbf{M}^{-1/2} \mathbf{M}^{-1/2}$$

Entonces

$$\mathbf{M}^{-1/2} \mathbf{M}^{-1/2} \mathbf{M} = \mathbf{M}^{-1/2} \mathbf{M} \mathbf{M}^{-1/2} = \mathbf{I}$$

En el caso que nos ocupa (el preblanqueo)

$$\mathbf{R}_{xx}^{-1/2} \mathbf{R}_{xx} \mathbf{R}_{xx}^{-1/2} = \mathbf{I}$$

Sin embargo, cuando se realiza preblanqueo se tiene

$$\mathbf{R}_{x'x'} = \mathbf{B} \mathbf{R}_{xx} \mathbf{B}^H$$

Esto no es un problema, puesto que

$$\begin{aligned} \mathbf{R}_{xx}^H &= \mathbf{R}_{xx} \\ \left(\mathbf{R}_{xx}^{-1/2}\right)^H &= \left(\mathbf{R}_{xx}^H\right)^{-1/2} = \mathbf{R}_{xx}^{-1/2} \end{aligned}$$

Por lo tanto, si

$$\mathbf{B} = \mathbf{R}_{xx}^{-1/2}$$

se tiene que

$$\mathbf{R}_{x'x'} = \mathbf{B} \mathbf{R}_{xx} \mathbf{B} = \mathbf{R}_{xx}^{-1/2} \mathbf{R}_{xx} \mathbf{R}_{xx}^{-1/2} = \mathbf{I}$$

Es decir, la matriz \mathbf{B} que acabamos de definir es una matriz de preblanqueo, alternativa a la utilizada por el análisis PCA visto en el capítulo 2. La opción 'sphering' (utilizada por defecto) emplea este método de preblanqueo que acabamos de ver.

La opción 'pca' sí realiza el preblanqueo mediante PCA. Sin embargo, sólo hace reducción (es decir, hay que eliminar, al menos, una componente principal), de manera que no puede ser utilizada para la realización del preblanqueo cuando queremos separar todas las componentes independientes.

Por eso la opción 'pca' sólo se puede utilizar en el caso en que el usuario elija reducir el número de componentes principales.

No obstante, la función runica permite, aunque no se esté realizando reducción con PCA y a pesar de que se trata de un algoritmo de separación, separar un número de componentes menor que el número total de observaciones.

Realmente lo que hace es realizar ICA sobre el vector de datos resultante de multiplicar un número de filas de la matriz de sphering por los datos originales. Es decir, si decidimos obtener L fuentes, runica realiza sphering y, posteriormente, la matriz weights de separación tendrá nulos los elementos correspondientes a las N-L últimas columnas.

Lo que se está haciendo es, pues, una reducción, pero no de componentes principales, ya que el sphering no se está realizando con PCA.

Esto impide que podamos hacer extracción y deflación utilizando la función runica, debido a lo siguiente: si hacemos extracción y deflación de las fuentes de 1 en 1 con preblanqueo antes de la primera extracción la matriz (vector) de extracción (weights) de ese primer paso constará de un primer elemento no nulo y el resto nulos. Esto hace que la matriz **P** de deflación tenga la primera fila y la primera columna nulas, y la submatriz restante sea la identidad. Esto significa que las nuevas observaciones tendrán una primera fila nula, y se producirá un error debido a que **R_{xx}** no será invertible.

La solución sería eliminar esa primera fila, con lo que las observaciones se reducen. Pero en ese caso lo que estamos haciendo es, en cada paso, obtener cada una de las observaciones preblanqueadas, algo que ya hemos hecho en el primer paso y que, además, no es análisis ICA. Veámoslo.

En el primer paso

$$\begin{aligned}
 \mathbf{y} &= \mathbf{Q}_1 \mathbf{B} \mathbf{x} = [q_{11} \ 0 \ 0 \ \dots \ 0] \mathbf{B} \mathbf{x} = [q_{11} \ 0 \ 0 \ \dots \ 0] \mathbf{x}' \\
 \mathbf{P} &= \mathbf{I} - \begin{bmatrix} 1/q_{11} \\ 0 \\ \vdots \\ 0 \end{bmatrix} [q_{11} \ 0 \ \dots \ 0] = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}
 \end{aligned}$$

Al realizar la deflación:

$$\mathbf{x}_D = \mathbf{P}\mathbf{x}'$$

Es decir

$$\mathbf{x}_D = \begin{bmatrix} 0 \\ x'_2 \\ x'_3 \\ \vdots \\ x'_N \end{bmatrix}$$

Eliminando el primer elemento

$$\mathbf{x}_D \leftarrow \begin{bmatrix} x'_2 \\ x'_3 \\ \vdots \\ x'_N \end{bmatrix}$$

Puesto que \mathbf{x}_D son señales preblanqueadas, en el segundo paso tendremos de nuevo una \mathbf{Q}_1 de la siguiente manera

$$\mathbf{Q}_1 = [q_{11} \quad 0 \quad \dots \quad 0]$$

Si en el primer paso no hacemos sphering tendremos una \mathbf{Q}_1 proporcional a la primera fila de la matriz \mathbf{B} resultante del sphering. Es decir, tendremos, paso a paso, los mismos resultados que haciendo sphering en el primer paso.

Continuando la descripción de `runicai5` sólo queda decir que, eligiendo uno de los tres algoritmos restantes, el programa introduce los datos en una variables llamada "x" y ejecuta el programa `bsseQN` (en el caso de `extract_QN`), `bsse2` (en el caso de `extract2`) o `fasticag` (en el caso de `fastica`).

En caso de elegir `fastica` los datos que se introducen en "x" (y que el usuario tendrá que cargar desde `fasticag`) ya son de media cero.

activi6

Éste es un programa de tan sólo dos líneas que genera las componentes independientes a partir de los datos, la matriz de sphering y la de separación. Tiene sentido en el caso de que se trabaje con matrices de separación y prewhitening obtenidas en análisis anteriores y almacenadas en archivos. Esto se hace para no almacenar las componentes separadas, ya que, al igual que los datos, ocupan mucho espacio en disco (sobre todo en MEG, donde hay una gran cantidad de sensores).

reordi7

Mediante este programa se reordenan las salidas (es decir, las componentes independientes). Se proponen dos opciones al usuario:

- según su contribución a las observaciones; en este caso se realiza la proyección de la componente en cada sensor y se calcula la media del cuadrado del valor en los sensores para cada instante de tiempo; después se elige el instante en que esa media cuadrática es máxima y ese valor será el que determine el orden
- según el máximo (eliminando la media) del valor cuadrático de las salidas; esto es útil para la representación gráfica, de manera que queden juntas señales con máximos absolutos parecidos y la escala de los ejes no deje invisible (por pequeña) a ninguna señal (hay que recordar que, para la representación gráfica, las señales se agrupan de 8 en 8), desde el punto de vista del análisis fisiológico este orden no tiene significado alguno, ya que el ICA proporciona las fuentes escaladas y no se sabe cuál es la escala.

Estas dos opciones se han implementado mediante dos funciones distintas. La primera opción es la que inicialmente proporciona el toolbox. Para conseguir la segunda sólo hay que modificar un par de líneas. Sin embargo, habría que utilizar otro parámetro y se ha preferido no modificar tanto la función original (**compsort**). Así, se han creado las funciones **compsort1** y **compsort2**. La función **compsort1** es para la primera opción. Hace exactamente lo mismo que **compsort** pero a una velocidad muchísimo mayor.

Además se ofrece (lo hace **compsort**) la posibilidad de reordenar las mayores componentes (el número de ellas lo elige el usuario; si es 0 no se realiza la nueva reordenación) en función de en qué instante de tiempo se produce el máximo (ya sea según el primer criterio o el segundo).

Las salidas generadas por las funciones **compsort1** o **compsort2** (y que, al ejecutar **reordi7**, quedan en el espacio de trabajo de MATLAB) son:

- **windex**: el vector que indica el orden de las componentes
- **maxvar**: el vector que contiene el valor máximo del criterio de ordenación para cada componente (ya con el nuevo orden dado por **windex**)

- `maxframe` y `maxepoch`: vectores que indican la posición del máximo (instante e intervalo) para cada componente (según el orden dado por `windex`)
- `maxmap`: matriz de proyección de las componentes ICA sobre los sensores; la columna *i*-ésima es la proyección de la componente *i*-ésima (según el nuevo orden establecido por `windex`) en el instante en que se produce su máximo según el criterio; esta matriz será utilizada para realizar el mapa cerebral de cada componente.

acploti8

Es prácticamente idéntico a `daploti2`, pero representa las componentes independientes (en el orden dado por `windex`) en vez de los datos.

prploti9

Este programa proyecta y representa el número de componentes solicitado por el usuario sobre los sensores que solicite también el usuario. Las componentes a proyectar son las más importantes (si el usuario solicita proyectar *L* componentes se proyectan las *L* mayores componentes según el orden de `windex`).

Además se muestra el porcentaje de varianza (`pvaf`) que corresponde a la suma de las proyecciones realizadas con respecto al total de varianza de los canales solicitados.

No se debe solicitar un número de canales muy elevado, porque ello ralentiza mucho la ejecución y dificulta la visión de las gráficas. Tampoco se debe utilizar un número elevado de componentes porque también ralentiza mucho el proceso.

La función utilizada para la proyección y la representación es **plotproj21**, que consiste en una mejora de la función **plotproj** perteneciente al toolbox para darle mayor velocidad. La diferencia está en que, antes de ejecutar `plotproj21`, `prploti9` calcula la matriz de mezclas (la inversa del producto de la matriz de separación por la de `prewhitening`), pasándosele a `plotproj21` como argumento. Así, esta última función calcula la proyección de cada componente como un producto de matrices, sin tener que llamar a **icaproj** (otra función del toolbox), como hace `plotproj`.

El porcentaje pvaf se calcula de la siguiente manera: se divide la varianza de las observaciones solicitadas (habiendo realizado primero la resta de las proyecciones) entre la varianza total de esas observaciones solicitadas. Esta cantidad se le resta a 1 y se obtiene pvaf.

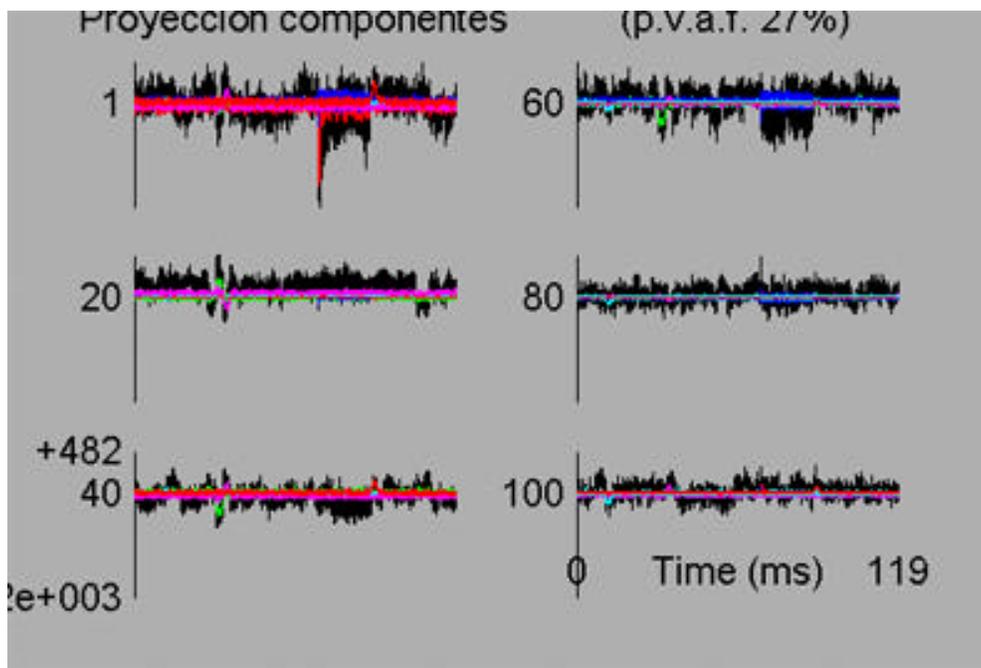


Figura 5-5 Ejemplo del uso de prploti9

entopi11

Dibuja la envolvente de los datos y las envolventes de las proyecciones de varias componentes solicitadas por el usuario. Además dibuja el mapa cerebral de cada componente indicando en qué instante se produce su máximo.

La envolvente de los datos son dos gráficas que encierran todos los datos. Es decir, una es el máximo de todos los datos para cada instante de tiempo y la otra el mínimo.

Lo mismo se puede decir para la envolvente de las proyecciones de una componente.

La función que realiza todo esto es **envtopo2**, que consiste en la función **envtopo** (del toolbox) en la que se ha eliminado el rótulo "potential (uV)".

En la figura 5-6 se puede ver una representación gráfica generada por el programa `entopi11`.

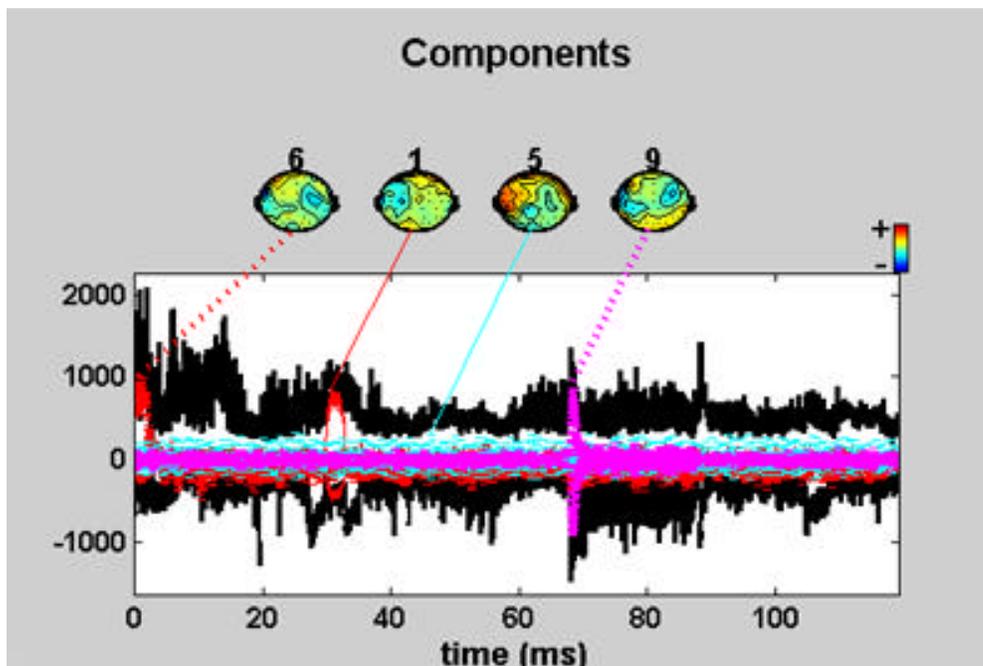


Figura 5-6 Ejemplo del uso de `entopi11`

`comapi12`

Dibuja los mapas cerebrales de las componentes deseadas por el usuario. Éste debe también indicar el número de filas y columnas que desea que tenga la representación. La función del toolbox utilizada para realizar la representación es `compmap`. En la figura 5-7 se muestra un ejemplo del uso de `comapi12`.

`icabpi13`

Dibuja un mapa cerebral de 1 sola componente y representa las gráficas de las proyecciones de esa componente en los 8 canales más importantes (es decir, los 8 canales en que esa componente se proyecta con más intensidad).

El usuario solicita la componente a representar.

La función utilizada es `compplot2` que es una mejora de `compplot` (función del toolbox). La función `compplot2` es más veloz. Además sólo representa 8 canales y 2000 muestras de cada uno, lo que permite visualizar correctamente las gráficas. Por otro lado corrige pequeños errores en la señalización del máximo en estas gráficas.

En la figura 5-8 se muestra un ejemplo del uso de icabpi13.

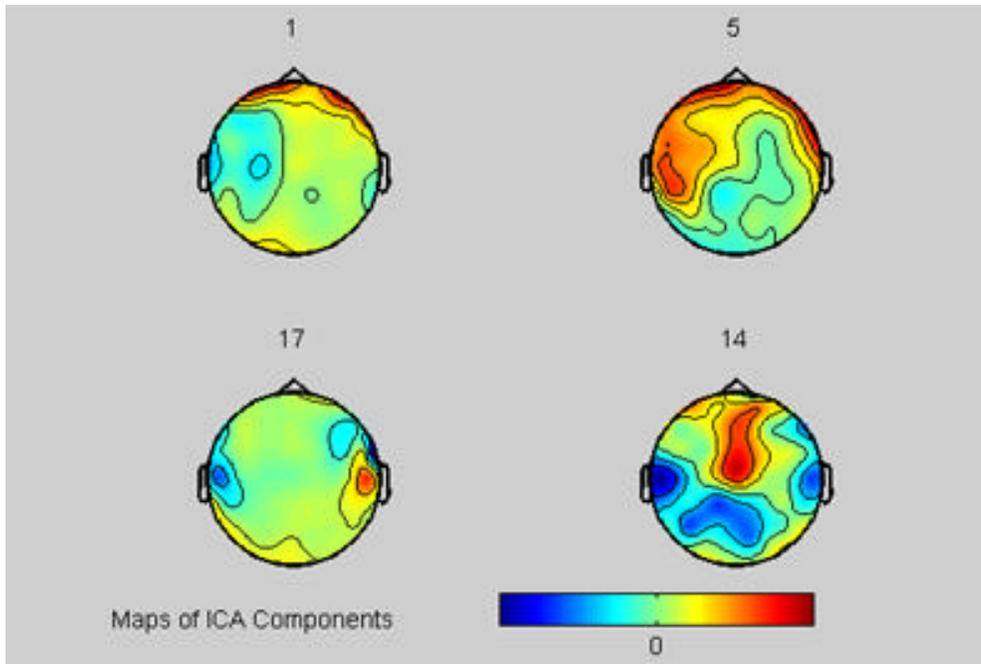


Figura 5-7 Utilización de comapi12

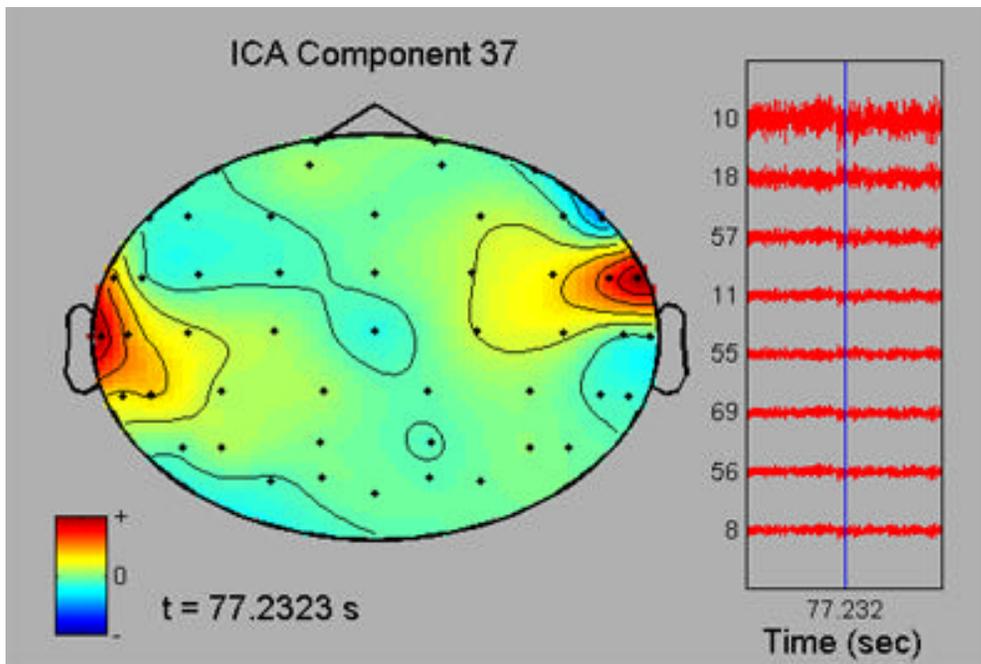


Figura 5-8 Uso de icabpi13

movi14

Similar a icabpi13. Dibuja un mapa cerebral, pero esta vez se trata de los propios datos. Realmente dibuja una secuencia de mapas cerebrales, que posteriormente (presionando el botón "movie") serán mostrados hacia delante y hacia atrás. El usuario decide las muestras (los instantes) en que se realizarán los mapas.

Cada mapa corresponde a los valores de las observaciones en un instante dado.

Paralelamente se representan las gráficas de los 8 sensores con niveles máximos dentro del intervalo en que se encuentran los instantes en que se realizarán los mapas. Dicho intervalo tiene un número de muestras solicitado por el usuario. En el centro de dicho intervalo se encuentra la secuencia de muestras para los mapas.

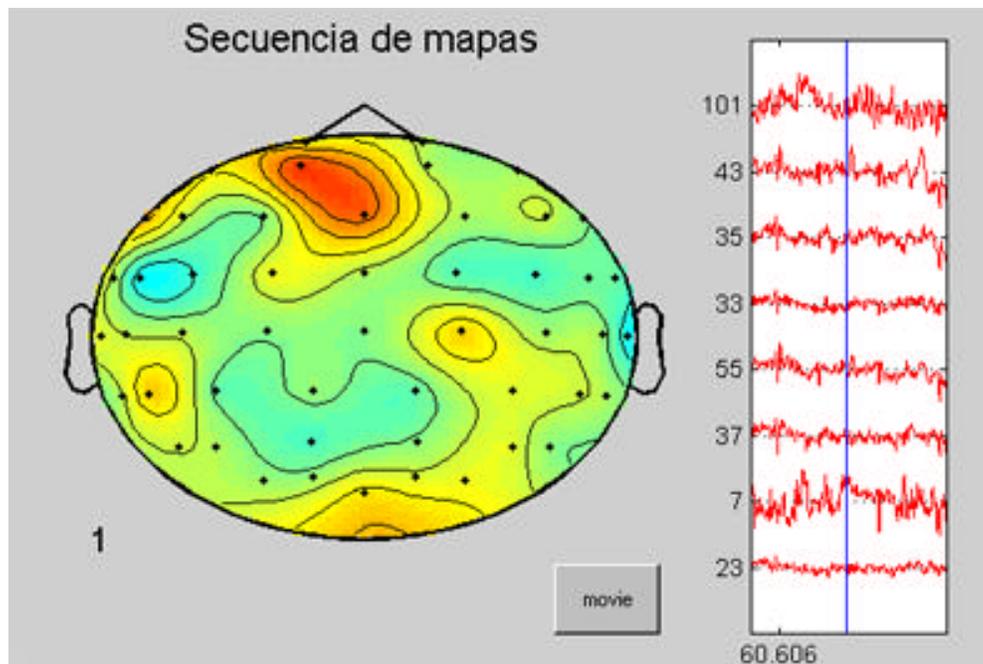


Figura 5-9 Primera muestra de la secuencia generada por movi14

Asociado a movi14 se ha realizado un pequeño programa de dos líneas, llamado **movim**, para la creación y utilización del botón que da comienzo al movimiento de la secuencia.

El mapa de colores toma como máximo el máximo absoluto de las observaciones en todo el intervalo, y como mínimo toma el opuesto de ese máximo absoluto.

La función que realiza los mapas y las gráficas es **eegmovie21**. Se trata de una modificación de **eegmovie** (la función que proporciona el toolbox). La modificación consiste únicamente en que se muestran sólo 8 canales (los que tienen mayores máximos absolutos) y no todos.

Para realizar el movimiento de los mapas, el programa `movim`, en la acción que se ejecuta cuando se hace clic en el botón hace una llamada a **seemovie**, función del toolbox.

5.4.3 Archivo de sensores 2-D

Para la ejecución de las funciones que realizan mapas cerebrales es necesario un archivo (*.loc) que localice los sensores en dos dimensiones (hay que recordar que los mapas son vistas cenitales del cerebro). Este archivo consta de 4 columnas. La primera (que es un entero) y la última (que son 4 caracteres) indican el número y el nombre del sensor, respectivamente.

La tercera columna indica el radio (la distancia del sensor al centro del dibujo). El valor máximo que puede tomar un elemento de esta columna es 0.5. Si toma un valor mayor se desechará ese sensor al hacer el mapa.

La segunda columna corresponde al ángulo (en grados) con respecto al eje de ordenadas. Se define el sistema de coordenadas bidimensional con origen en el centro del cerebro, y se considera la cabeza vista desde arriba con la cara en la parte superior de la pantalla. El ángulo es positivo hacia la derecha.

Por lo tanto, si llamamos L_{i2} y L_{i3} a los elementos de la segunda y tercera columna correspondientes al sensor i tenemos que

$$x_i = L_{3i} \sin\left(L_{2i} \frac{\pi}{180}\right)$$

$$y_i = L_{3i} \cos\left(L_{2i} \frac{\pi}{180}\right)$$

son las coordenadas de dicho sensor en el sistema que acabamos de definir.

5.5 CONCLUSIONES

Se han descrito en este capítulo las características más importantes de los programas para MATLAB que van a ser utilizados en el capítulo siguiente de simulaciones.

Los tres algoritmos de extracción (`extract_QN`, `extract2` y `fastica`) ofrecen diversas posibilidades en la manera de extraer y la cantidad de extracciones (desde extraer todas las componentes de una vez hasta extraerlas de una en una, o bien extraer sólo un conjunto de componentes, bien de una vez o bien por grupos).

Tanto `fastica` como `runica` ofrecen la posibilidad de utilizar el algoritmo por defecto u otras variantes. Los otros dos algoritmos (tal como están implementados) no proporcionan esa posibilidad, aunque pueden ser fácilmente modificados para que realicen las variaciones que se han descrito en este capítulo.

El interfaz gráfico que posee `fastica` es muy cómodo de utilizar para la realización del análisis ICA, aunque no posee funciones de representación de datos y salidas que permitan al usuario visualizar los resultados cómodamente.

El toolbox MATLAB ICA/EEG es una herramienta de gran utilidad, que en este proyecto no ha sido empleada con todas sus posibilidades.

Tanto las funciones que permiten representar las gráficas de las componentes como aquellas que permiten la realización de mapas cerebrales a partir de las proyecciones de esas componentes en los sensores serán empleadas para identificar los distintos artefactos que tendrán que ser eliminados.

CAPÍTULO 6

SIMULACIONES

En este capítulo se analizará el comportamiento de los distintos algoritmos (y sus variantes) estudiados en los capítulos anteriores.

Para ello se va a emplear un archivo de datos procedentes de un MEG. A estas observaciones se les aplicarán los diversos algoritmos ICA de manera que se podrá comprobar la efectividad de cada uno de ellos al separar (o extraer) los diferentes tipos de señales.

Compararemos la calidad de las señales obtenidas (según su parecido con las señales ya conocidas), la velocidad con que se ejecutan los algoritmos y que componentes son extraídas en primer lugar.

Por otro lado veremos qué ocurre cuando se reduce el número de componentes principales. Se trata de ver qué componentes independientes son capaces de extraer los algoritmos a partir de un determinado número de componentes principales (las de mayores valores absolutos de sus autovalores).

Por último, para cada algoritmo, se generarán las señales definitivas que deben ser entregadas al profesional (médico, fisiólogo, psiquiatra, etc.) para su análisis. Es decir, a los datos se les hará deflación de los artefactos detectados.

Para la realización de las simulaciones se empleó un PC con procesador AMD K6-2 a 350 MHz y 64 Mb de RAM.

6.1 OBSERVACIONES

Las observaciones consisten en un archivo de datos ASCII obtenidos en [2] y que contiene las señales recibidas por los sensores de un neuromagnetómetro. Al cargar

el archivo aparece una variable (MEG_art) que contiene los datos. Por razones de velocidad y de espacio en disco duro se ha almacenado esta variable en un archivo con formato de MATLAB llamado megart.mat.

Los datos proceden del experimento realizado en [Vigário97]. Se trata de las señales obtenidas en un MEG realizado en el neuromagnetómetro Neuromag-122, que tiene 122 sensores en 61 puntos diferentes de la superficie de la cabeza del paciente, de manera que en cada punto hay 2 sensores ortogonales.

Para producir los artefactos típicos que deben ser posteriormente detectados en el análisis ICA y eliminados, el sujeto sobre el que se realizó el experimento realizó movimientos horizontales de ojos y parpadeos. También apretó los dientes durante 20 segundos para producir artefactos miográficos.

Otro artefacto fue creado colocando un reloj digital a 1 metro de distancia de la cabeza del sujeto en la zona posterior derecha.

Al lado del ombligo se le colocó una pieza metálica para hacer más fuerte el artefacto producido por la respiración.

Por lo tanto, los artefactos que deberán ser encontrados por los algoritmos son:

- VEOG



Figura 6-1 Parte de la gráfica de un artefacto correspondiente a VEOG

- HEOG



Figura 6-2 Parte de la gráfica de un artefacto correspondiente a HEOG

- ECG



Figura 6-3 Parte de la gráfica de un artefacto correspondiente a ECG

- Reloj digital

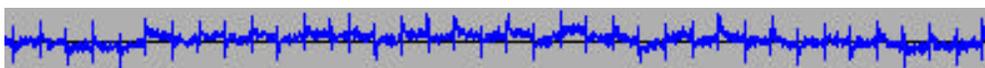


Figura 6-4 Parte de la gráfica de un artefacto correspondiente al reloj

- Mordedura

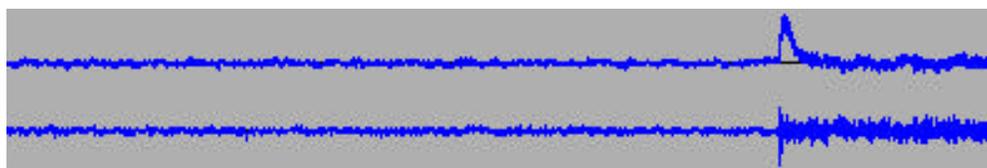


Figura 6-5 Parte de las gráficas de dos artefactos provocados por la mordedura

- Respiración



Figura 6-6 Parte de la gráfica de un artefacto debido a la respiración

A pesar del parecido de las gráficas del ECG y del reloj digital, las diferencias son visibles. En la figura 6-7 se muestran varios de los pulsos del artefacto debido al reloj. Cuando la curva tiene un ascenso brusco, después cae lentamente, y viceversa.

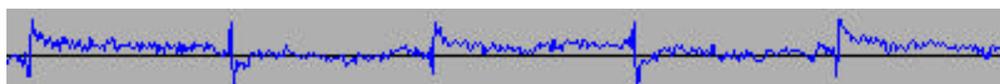


Figura 6-7 Detalle de la gráfica del artefacto correspondiente al reloj

Los datos fueron filtrados con la banda de paso [0.03 - 90 Hz]. Posteriormente fueron digitalizados con una frecuencia de muestreo de 297 Hz. Por último, fueron filtrados digitalmente con un filtro LP de frecuencia de corte 45 Hz y submuestreados por un factor de 2. Por tanto, la frecuencia máxima de los datos es de 45 Hz.

La variable de los datos es, pues, una matriz de 122 filas (los sensores) y 17730 columnas (las muestras). La duración de la grabación de los datos es de, aproximadamente, 2 minutos.

Para facilitar la identificación visual de las distintas componentes independientes se realizó a la vez un ECG, un VEOG y un HEOG.

Para la realización de los mapas cerebrales se utilizará el archivo megarloc.loc. Dicho archivo contiene las posiciones bidimensionales de los sensores (tal como se ha descrito en el capítulo anterior). El archivo original, que contenía las localizaciones de los sensores fue enviado por L. Parkkonen [3].

Hubo que realizar un cambio de coordenadas (de cartesianas a las polares utilizadas por el toolbox) para conseguir el archivo megarloc.loc.

Existe además otro pequeño problema. Debido a que el toolbox está pensado para trabajar con datos de EEG, aparece como aviso el hecho de que dos sensores estén localizados en el mismo punto.

Sin embargo, el toolbox realiza una media de los dos valores. Esto resuelve el problema, ya que la magnitud de los datos de un MEG es el flujo magnético. Los sensores son ortogonales y de la misma superficie, por lo que el valor total de cada punto es la suma de los valores recogidos por ambos sensores. No importa que el toolbox realice una media y no una suma, ya que eso supone dividir por 2 el valor de los 61 puntos, lo cual no afecta al mapa cerebral.

Se podría haber creado un fichero *.loc de 61 posiciones y sumar los datos para introducirlos en las funciones que representan los mapas cerebrales, pero, como en varias de ellas se utilizan también las representaciones de las gráficas de los datos, se ha optado por dejar que se realicen las medias.

Un último problema es que algunos de los sensores (correspondientes a 11 puntos) quedan, según la coordenada z, por debajo de los puntos de la cabeza que es capaz de representar el toolbox. En el análisis ICA intervienen todos los sensores, pero a los 11 puntos que quedarían en la parte inferior trasera de la cabeza (y provocarían confusión, ya que son puntos que no se pueden ver desde un plano cenital) se les ha modificado el radio hasta un valor de 5, de manera que el toolbox los desprecia a la hora de realizar el mapa cerebral.

6.2 COMPARACIÓN DE ALGORITMOS

Primero vamos a describir las distintas simulaciones que se han llevado a cabo, así como los resultados obtenidos, para, posteriormente, realizar una comparación de estos resultados.

Como se describirá, algunas de las componentes (VEOG, HEOG y, sobre todo el reloj digital) no son correctamente separadas o extraídas (o bien no aparecen o bien aparecen mezcladas y ocultas por otras componentes) en ninguno de los algoritmos. Esto se debe a la gran cantidad de componentes introducidas por las respiraciones. Para

evitar estas componentes impidan la correcta obtención de otras se realizaron segundas simulaciones tras filtrar los datos con un filtro HP con frecuencia de corte 1 Hz. Esto elimina las componentes debidas a las respiraciones.

6.2.1 runica I

En este primer caso utilizamos:

- algoritmo runica INFOMAX
- sin reducción de componentes principales
- separación de todas las componentes independientes
- valor de stop 10^{-8}

El algoritmo se ejecutó en un tiempo de 1 hora y 30 minutos.

El resultado es:

- componente 1: movimiento horizontal (no bien definido), aunque un poco enmascarado por un movimiento de respiración
- componente 2: posible parpadeo de ojos (VEOG) enmascarado por respiración
- componente 3: probable superposición de movimiento horizontal (HEOG) de ojos y respiración
- componente 5: ECG
- componentes 9, 15, 17, 27, 36, 37, 69, 88: tensión muscular por presión de mandíbulas (mordedura)
- componentes 22, 97, 102, 109, 117, 118: tienen una componente que parece ser el reloj (según los mapas cerebrales), pero muy oculta por otras componentes no identificadas como artefactos
- componentes 24, 46, 47, 59, 73, 76, 82, 89, 95-98, 100, 102, 103, 106, 108-113, 115, 117-122: componentes no identificadas como artefactos
- componentes 4, 99, 101, 104, 105, 107, 114, 116: superposiciones de respiraciones y otras componentes
- resto de componentes: respiraciones

En general los resultados no son demasiado buenos, siendo la componente del ECG, las componentes debidas a mordeduras y las debidas a respiraciones las únicas que están claramente separadas. Muchos artefactos quedan, pues, ocultos en los datos.

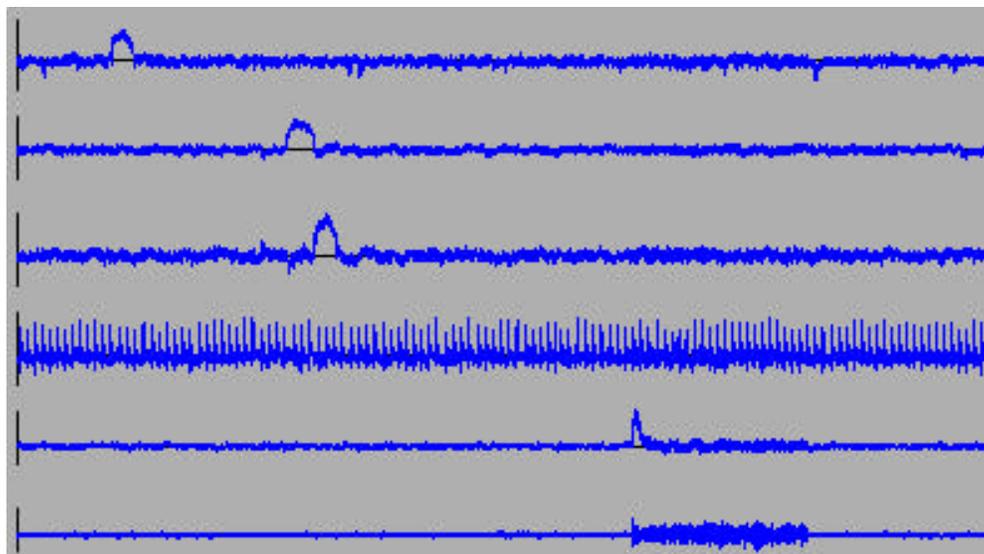


Figura 6-8 Gráficas de las componentes 2, 1, 3, 5, 9 y 15

Se puede comprobar que las componentes debidas a EOG (tanto VEOG como HEOG) y la debida al reloj prácticamente no se aprecian (sobre todo el reloj y el VEOG). El reloj se descubre en algunas componentes gracias a los mapas cerebrales (como ocurre en la componente 22).

El problema es que es mayor el número de componentes que el de observaciones, debido, fundamentalmente, a la gran cantidad de componentes introducidas por la respiración.

Esto se resuelve de la siguiente manera: las componentes debidas a la respiración son señales independientes cuyas formas de ondas son pulsos suavizados (no cuadrados) de, aproximadamente, 500 muestras. Esto significa que su espectro tiene una frecuencia de caída de 3 dB de, aproximadamente, $1/(\pi \cdot 500) \approx 0.1$ Hz.

Por otra parte, si consideramos la señal asociada a respiración como una sola señal que contiene todos los pulsos, la envolvente del espectro tendrá igualmente una caída de 3 dB en 0.1 Hz, aproximadaente.

Por lo tanto, para eliminar las componentes debidas a la respiración se filtró la variable de datos con un filtro HP de frecuencia de corte 1 Hz (tal como se muestra en [Vigário97]). Como se verá, esto permitió encontrar VEOG, HEOG y el reloj digital.

El procedimiento llevado a cabo es, pues: filtrado HP de 1 Hz, tras la carga de los datos; posteriormente se realiza el análisis ICA; se almacenan las matrices de separación y prewhitening; por último, para visualizar los resultados (las componentes independientes) se multiplican las matrices de separación y prewhitening por los datos (sin media) originales.

No debe extrañar esto último. Al filtrar \mathbf{x} obtenemos \mathbf{x}^F . Al realizar el ICA a \mathbf{x}^F obtenemos $\mathbf{y}_F = \mathbf{W}_F \mathbf{x}^F$. El mismo resultado se obtiene si aplicamos primero la matriz del análisis ICA ($\mathbf{y} = \mathbf{W}_F \mathbf{x}$) y luego filtramos, de modo que $\mathbf{y}^F = (\mathbf{W}_F \mathbf{x})^F = \mathbf{W}_F \mathbf{x}^F$. Esto se debe a que el filtro es LTI, y cada y_i es una combinación lineal de las x_i .

Como pretendemos conocer las componentes \mathbf{y} que se hallan en \mathbf{x} , evitamos el último paso del segundo procedimiento que acabamos de ver (no filtramos \mathbf{y}), de modo que se trata solamente de aplicar la matriz de separación, \mathbf{W}_F , (obtenida del análisis ICA sobre los datos filtrados) a los datos originales.

Como resultado obtenemos las componentes del VEOG (la número 1), del HEOG (la número 10) y del reloj (la número 9).

Para limpiar las observaciones habrá que eliminar unos artefactos empleando una matriz de separación (la obtenida del análisis de los datos originales, para el caso del ECG, la mordedura y la respiración) y otros artefactos empleando la otra matriz (la obtenida del análisis sobre los datos filtrados, para el caso del VEOG, el HEOG y el reloj).

El tiempo de análisis para obtener la separación a partir de los datos filtrados fue de 1 hora y 50 minutos.

6.2.2 runica II

En este caso se utilizó:

- algoritmo runica extendido estimando el signo cada 10 bloques
- sin reducción
- separación de todas las componentes

- valor de stop 10^{-6}

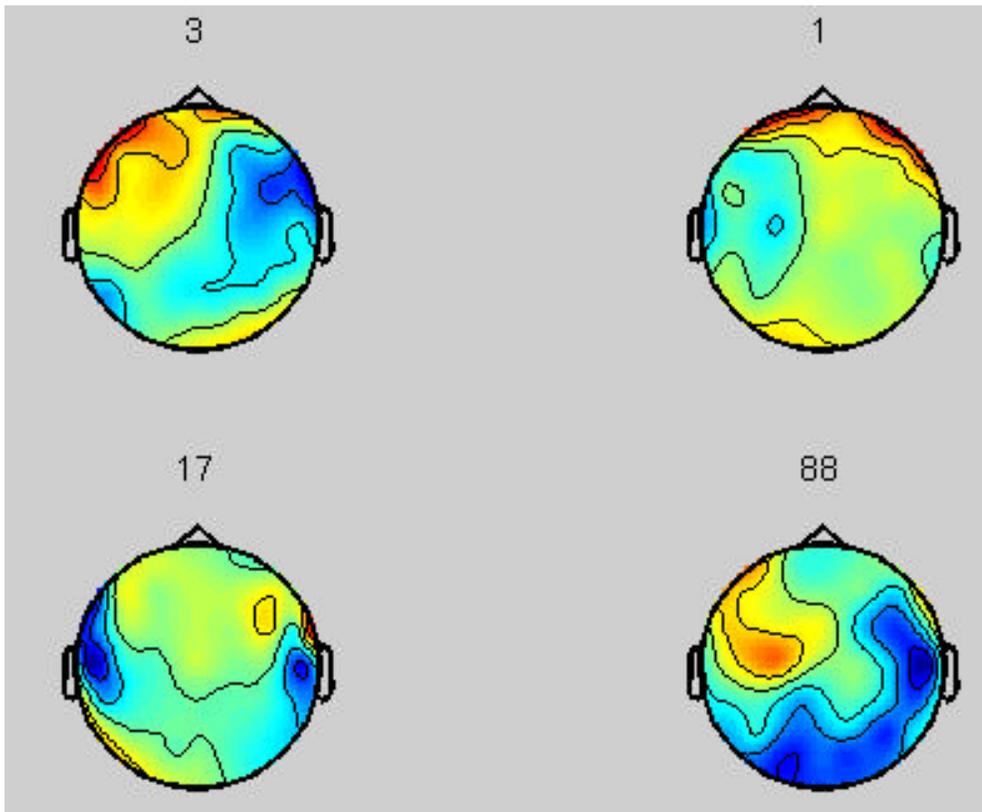


Figura 6-9 Mapas de algunas componentes obtenidas antes de filtrar

El tiempo de análisis fue de 10 horas

Los resultados fueron:

- componentes 1 y 2: HEOG superpuesto a respiraciones probablemente
- componentes 3 y 63: VEOG superpuesto a respiraciones probablemente
- componente 4: ECG
- componentes 79, 82, 83, 88 y 94: probablemente reloj superpuesto a otras señales (en la 88 hay una componente asociada a mordedura)
- componentes 9, 17, 23, 40, 43, 44, 67, 84 y 110: mordedura (en las tres últimas superpuesta a otras señales)
- componentes 106, 108, 109: respiraciones superpuestas a otras componentes

- componentes 32, 34, 85, 89, 91, 93, 95-97, 98-105, 107, 111-122: otras componentes no identificadas como artefactos
- resto de componentes: respiraciones.

Nuevamente lo único que aparece claro es el ECG, las mordeduras y las respiraciones. La gran cantidad de salidas relacionadas con estas últimas impide separar los artefactos que nos interesa eliminar.

De nuevo filtramos con filtro HP de 1 Hz, eliminando las componentes debidas a respiraciones.

El resultado es la obtención del VEOG en la componente 1, el HEOG en la 8 y el reloj digital en la componente 7, con un tiempo de ejecución fue de 9 horas y 20 minutos.

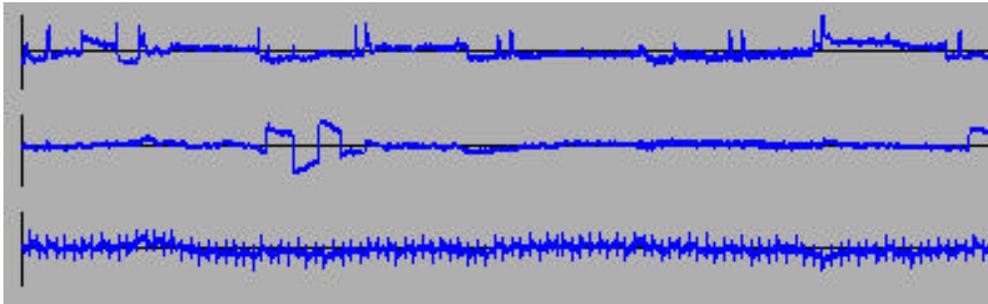


Figura 6-10 Gráficas de las componentes 1, 8 y 7 obtenidas tras filtrar

6.2.3 fastica I

Para este análisis se utilizó:

- algoritmo fastica con separación, 'pow3' y estabilización
- sin reducción
- separación de todas las componentes
- valor de stop 10^{-6}

El algoritmo convergió en 2 horas.

Los resultados fueron:

- componente 61: ECG

- componentes 2, 43, 60, 89, 108, 112: mordedura
- componentes 11, 59: mordedura superpuesta a respiraciones
- componentes 84, 118: mordedura superpuesta a otras componentes no identificadas como artefactos
- componentes 29, 35, 39, 73, 75, 85, 94, 96, 99, 105: componentes no identificadas como artefactos
- resto de componentes: respiraciones

La separación no es buena. Quedan totalmente ocultos los EOG y el reloj. Hay poca separación entre los artefactos encontrados y demasiadas componentes de respiraciones.

Al filtrar con filtro HP 1 Hz obtenemos VEOG (70), HEOG (121) y el reloj (33). Sin embargo, el algoritmo no llegó al final (al valor de stop): tras 12 horas llegó a 500 iteraciones, deteniéndose.

6.2.4 fastica II

Para este análisis se utilizó:

- algoritmo fastica con deflación, 'pow3' y estabilización
- sin reducción
- extracción de todas las componentes
- valor de stop 10^{-5}

Se subió el valor de stop porque mientras el valor por defecto que tiene runica es 10^{-6} , el que tiene fastica es 10^{-4} .

El algoritmo se ejecutó en 2 horas y 30 minutos.

Los resultados fueron:

- componentes 3, 11, 12, 29, 38: HEOG muy oculto por respiraciones
- componente 62: ECG
- componentes 99, 110, 111, 112: reloj muy camuflado por otras componentes
- componentes 1, 16, 26, 52, 69: mordedura

- componentes 20, 28, 34, 51, 59, 63, 71, 93, 96: mordedura superpuesta a respiraciones y otras componentes
- componentes 81, 82, 86-92, 97, 98, 104: respiraciones superpuestas a otras componentes
- componentes 100-103, 105-122: componentes no identificadas como artefactos
- resto de componentes: respiraciones

Otra vez las respiraciones impiden una correcta separación de los artefactos.

Al filtrar con HP 1 Hz obtuvimos VEOG (4), HEOG (11) y reloj (9), con un tiempo de ejecución de 5 horas y 45 minutos.

6.2.5 fastica III

Se utilizaron las siguientes opciones:

- algoritmo fastica con deflación, 'tanh', y estabilización
- sin reducción
- extracción de todas las componentes
- valor de stop 10^{-5}

La ejecución duró 9 horas y 30 minutos.

Los resultados fueron:

- componentes 5, 11, 21: HEOG poco claro (la componente 5 es la que más se acerca a la forma de onda del HEOG)
- componente 47: ECG
- componentes 1, 22, 30, 32, 34, 50, 51, 63, 79: mordedura
- componente 29: HEOG y respiración
- componente 68: mordedura superpuesta a respiraciones
- componente 80: mordedura superpuesta a reloj
- componentes 81, 85: respiraciones superpuestas reloj

- componentes 71, 72, 83, 84, 86, 87, 89-91, 95, 98, 99: respiraciones superpuestas a otras componentes
- componentes 70, 88, 92-94, 96, 97, 100-122: no identificadas como artefactos
- resto de componentes: respiraciones

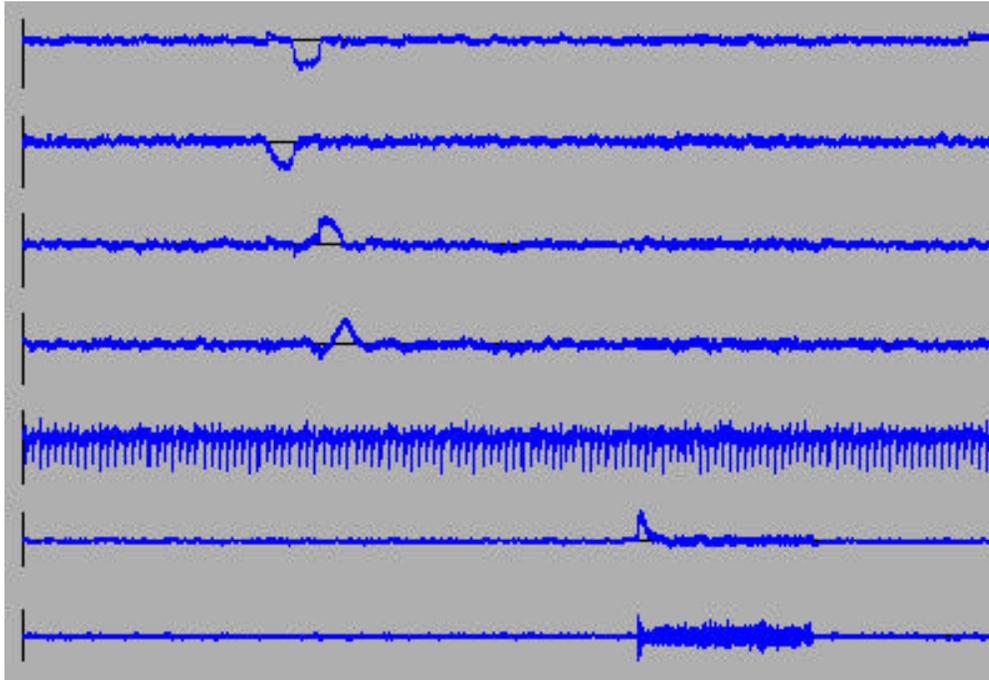


Figura 6-11 Gráficas de las componentes 5, 11, 21, 29, 47, 1 y 22

Hay menos componentes debidas a respiración porque algunas de ellas aparecen superpuestas en la misma salida (salidas con más de un pulso de respiración).

De nuevo las únicas componentes claramente identificadas son: ECG, mordedura y respiraciones. Sin embargo esta vez el HEOG es más visible.

Filtrando con filtro HP 1 Hz se obtuvo VEOG (4), HEOG (10) y reloj(23), con un tiempo de ejecución de 17 horas y 30 minutos.

6.2.6 fastica IV

Las opciones utilizadas fueron:

- algoritmo fastica con deflación, 'gauss', y estabilización
- sin reducción

- extracción de todas las componentes
- valor de stop 10^{-5}

La ejecución duró 8 horas y 30 minutos.

Los resultados fueron:

- componentes 29, 51: superposición de VEOG y respiración
- componentes 4, 14: HEOG (en el caso de la componente 4 bastante claro, en el de la 14 no tanto)
- componente 49: ECG
- componentes 1, 15, 23, 27, 31, 46, 50: mordedura
- componentes 12, 24, 42, 51, 58, 59: superposición de mordedura y respiraciones
- componentes 73, 80: superposición de mordedura y otras señales
- componentes 41, 43: reloj muy oculto por respiraciones
- componente 66: reloj muy oculto por otras señales
- componentes 60, 61, 81-87: superposición de respiraciones y otras componentes
- componentes 65, 75, 77-79, 88-122: componentes no identificadas como artefactos
- resto de componentes: respiraciones.

Las componentes claramente identificadas fueron el ECG, la mordedura, las respiraciones y, en menor medida, el HEOG. Aparecen algunas componentes debidas a respiración unidas en la misma componente.

Se realizó filtrado HP de 1 Hz y se obtuvo el VEOG (4), el HEOG (10) y el reloj (18).

La ejecución duró 11 horas y 30 minutos.

6.2.7 fastica V

Las opciones utilizadas fueron:

- algoritmo fastica con deflación, 'skew' y estabilización
- sin reducción
- extracción de todas las componentes
- valor de stop 10^{-5}

La ejecución duró 6 horas.

Los resultados fueron:

- componentes 9, 12: HEOG (la 12 más clara)
- componente 51: ECG
- componentes 1, 67, 79, 88: mordedura (1 y 67 las más claras)
- componentes 84, 85, 89, 96: superposición de mordedura y otras componentes
- componentes 90, 94: superposición de mordedura, respiración y otras componentes
- componentes 81-83, 86, 87, 92, 93: superposición de respiraciones y otras componentes
- componentes 91, 95, 97-122: componentes no identificadas como artefactos
- resto de componentes: respiraciones.

Aparecen claramente el ECG, el HEOG (no demasiado), las respiraciones y sólo dos salidas relacionadas con la mordedura. La mordedura aparece superpuesta en una gran cantidad de componentes.

Se realizó filtrado HP 1Hz y se encontraron VEOG (1), HEOG (11, 32), reloj (14, no demasiado claro) y mordeduras (3-5, 9, 10).

La ejecución duró 18 horas.

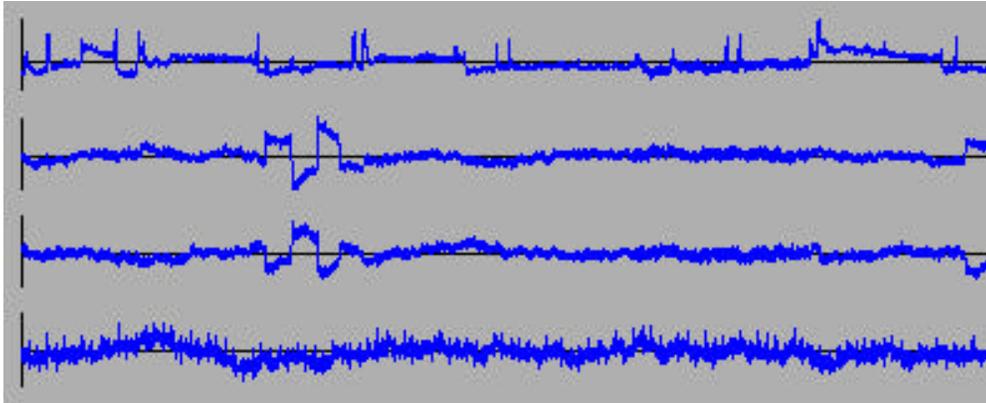


Figura 6-12 Gráficas de las componentes 1, 11, 32 y 14, obtenidas tras el filtrado de los datos

6.2.8 extract_QN

Se utilizaron las opciones:

- algoritmo extract_QN
- sin reducción
- extracción de todas las componentes
- valor de stop 10^{-5}

La ejecución duró 6 horas.

Los resultados fueron:

- componentes 1, 14: HEOG
- componente 8: ECG
- componentes 15, 18, 32, 56, 68, 73: mordedura
- componente 36: mordedura superpuesta a respiraciones
- componente 93: superposición de mordedura y otras componentes
- componente 89: posiblemente reloj y otras componentes (por la forma del mapa)
- componentes 39, 67, 80, 83, 90-92, 94, 96-103: respiraciones superpuestas a otras componentes

- componentes 2, 6, 11, 24, 38, 58, 95, 104-122: componentes no identificadas como artefactos
- resto de componentes: respiraciones

Se muestran claramente el ECG, el HEOG (no demasiado), las respiraciones y la mordedura. Las componentes 2, 6, 11, 24, 38 presentan una forma extraña.

Realizando filtrado HP 1 Hz se obtuvieron VEOG (2), HEOG (17) y reloj (27) con un tiempo de ejecución de 6 horas y 30 minutos.

6.2.9 extract2

Las opciones utilizadas fueron:

- algoritmo extract2
- sin reducción
- extracción de todas las componentes
- valor de stop 10 - 5

La ejecución duró 9 horas.

Los resultados fueron:

- componente 66: respiración y (probablemente) VEOG
- componentes 1, 2: HEOG
- componente 11: ECG
- componentes 4, 7, 18, 36, 60-62, 75: mordedura
- componentes 91, 93: superposición de mordedura y otras componentes
- componentes 87, 89, 90: superposición de respiración, posiblemente reloj y otras componentes
- componentes 76, 82-86, 88, 92, 94-106: superposición de respiración y otras componentes
- componentes 9, 107-122: componentes no identificadas como artefactos
- resto de componentes: respiraciones

Se separaron claramente el ECG, el HEOG (no demasiado), la mordedura y las respiraciones.

La componente 9 presenta una forma extraña.

Se filtró con HP 1 Hz y se obtuvieron el VEOG (3), el HEOG (6) y el reloj (10).

La ejecución duró 13 horas.

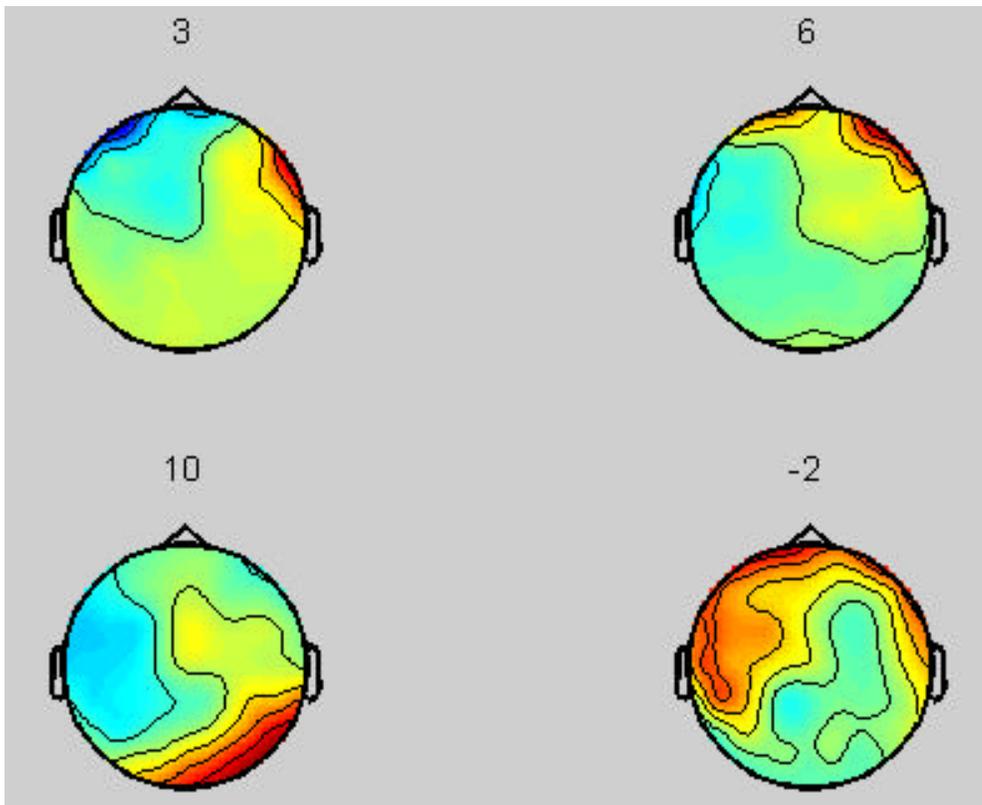


Figura 6-13 Mapas de algunas componentes obtenidas tras filtrar

6.2.10 runica III

Las opciones utilizadas fueron:

- algoritmo runica INFOMAX
- reducción a 15 componentes principales
- valor de stop 10^{-8}

La ejecución duró 4 minutos y 45 segundos.

Los resultados fueron:

- componente 2: VEOG y mordedura
- componente 1: HEOG
- componente 13: ECG
- componentes 4, 8: mordedura
- componente 7, 12: superposición de mordedura y reloj
- componente 11: superposición de mordedura y respiración
- componente 3: superposición de mordedura, reloj (muy poco) y respiración
- componentes 5, 9: respiraciones
- componentes 6, 10, 14, 15: otras componentes no identificadas como artefactos.

Se muestran con claridad VEOG, HEOG, ECG y mordedura.

Tras filtrar con filtro HP 1 Hz y un análisis de 5 minutos y 10 segundos:

- componente 1: VEOG
- componente 14: HEOG
- componentes 5, 15: superposición de HEOG (poco) y mordedura
- componente 2: ECG
- componentes 3, 8: mordeduras
- componente 9: reloj no muy claro
- resto de componentes: componentes no identificadas como artefactos.

El reloj aparece, aunque no tan claramente como en los casos en que no hay reducción de componentes principales.

6.2.11 runica IV

Las opciones utilizadas fueron:

- algoritmo runica extendido calculando el signo cada bloque
- reducción a 15 componentes principales
- valor de stop 10^{-8}

La ejecución duró 35 minutos y 30 segundos.

Los resultados fueron:

- componente 3: superposición de VEOG y ECG (domina el VEOG)
- componente 6: superposición de VEOG y mordedura
- componente 2: HEOG
- componente 10: ECG
- componentes 4, 9 11, 12, 15: mordedura
- componente 5: superposición de mordedura y otras componentes
- componente 8: respiración
- resto de componentes: no identificadas como artefactos.

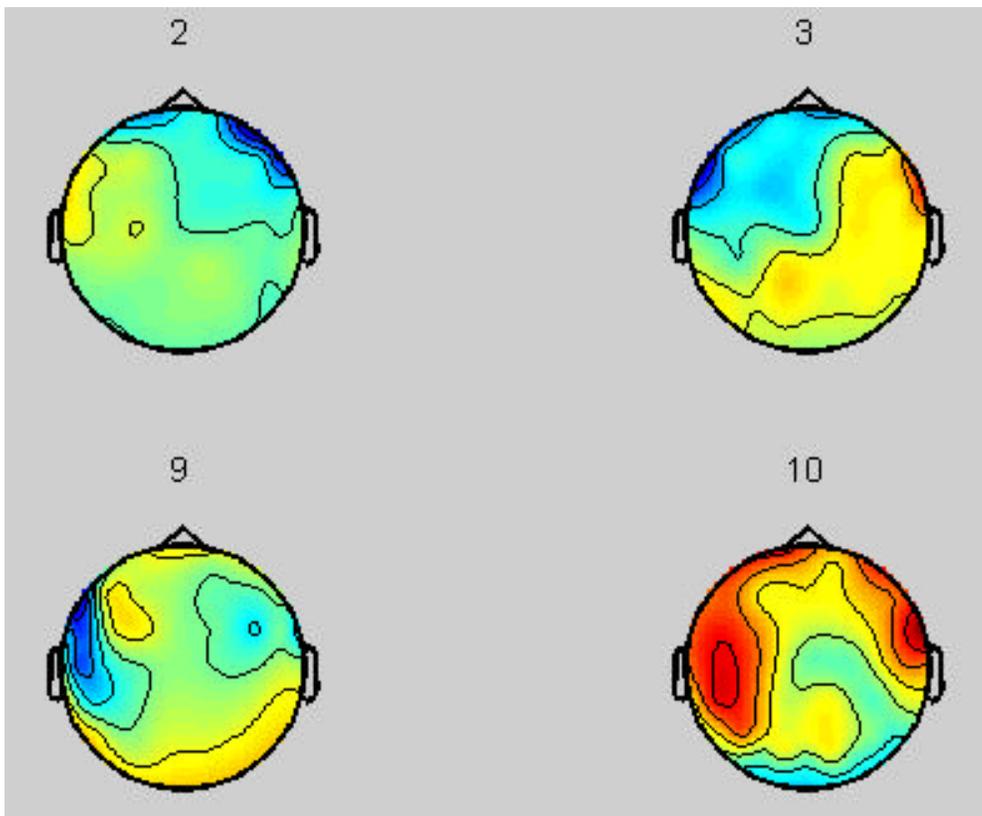


Figura 6-14 Mapas de algunas componentes

Las componentes que aparecen más claras son el VEOG (no demasiado, por culpa del ECG), el HEOG, el ECG, la mordedura y una respiración.

Al filtrar con HP 1 Hz se obtuvo:

- componente 1: VEOG
- componente 15: HEOG
- componentes: 3, 11: HEOG superpuesto a mordedura
- componente 2: ECG
- componentes 5, 9: mordedura
- componente 14: superposición de mordedura y otras componentes

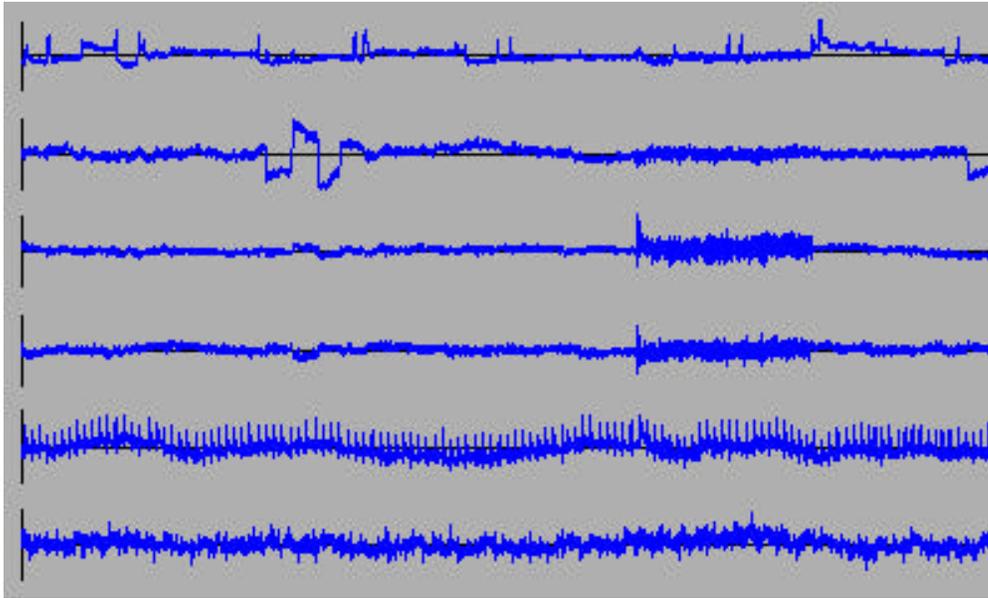


Figura 6-15 Gráficas de las componentes 1, 15, 3, 11, 2 y 8 obtenidas tras el filtrado de los datos

- componente 8: reloj (no demasiado claro)
- resto de componentes: no identificadas como artefactos.

Se obtiene el VEOG de forma mucho más clara. El reloj aparece de manera más visible, pero de todas formas no es muy claro.

La ejecución duró 32 minutos y 30 segundos.

6.2.12 fastica VI

Las opciones utilizadas fueron:

- algoritmo fastica con deflación, 'pow3' y estabilización

- reducción a 15 componentes principales
- valor de stop 10^{-6}

El tiempo de ejecución fue de 8 minutos.

Los resultados fueron:

- componente 4: superposición de VEOG (dominante) y ECG (muy poco)
- componentes 2, 5: superposición de VEOG y mordedura
- componente 8: HEOG
- componente 6: ECG
- componentes 1, 3, 7: mordedura
- componente: 10: mordedura y otras componentes
- componente 15: reloj (poca intensidad) y otras componentes
- componente 11: respiración
- resto: componentes no identificadas como artefactos.

Lo más claro es el VEOG (no demasiado), el HEOG, el ECG, la mordedura y una respiración.

Al filtrar con filtro HP 1 Hz se obtuvieron los siguientes resultados:

- componente 2: VEOG
- componentes 3, 4, 5, 7: HEOG superpuesto a mordedura
- componente 8: HEOG
- componente 6: ECG
- componente 1: mordedura
- componente 9: reloj (no demasiado claro)
- resto: componentes no identificadas como artefactos.

Se obtiene de forma clara el VEOG. El reloj no aparece con tanta claridad.

El tiempo de ejecución fue de 5 minutos y 50 segundos.

6.2.13 fastica VII

Las opciones elegidas fueron:

- algoritmo fastica con deflación, 'tanh' y estabilización
- reducción a 15 componentes principales
- valor de stop 10^{-6}

Tras 6 minutos y 30 segundos de ejecución los resultados fueron:

- componente 2: VEOG (dominante) superpuesta al ECG
- componente 6: HEOG
- componente 8: ECG
- componentes 1, 5, 7, 9: mordedura
- componente 4: mordedura y respiración

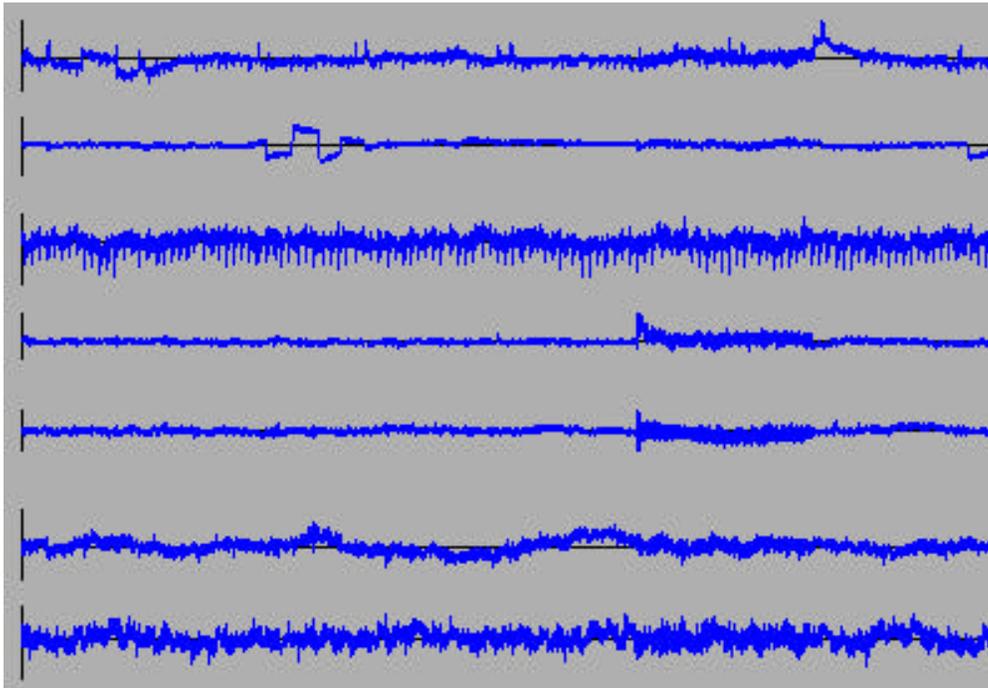


Figura 6-16 Componentes 2, 6, 8, 1, 5, 11 y 15

- componente 11, 15: reloj (poco) superpuesto a otras componentes
- resto: componentes no identificadas como artefactos

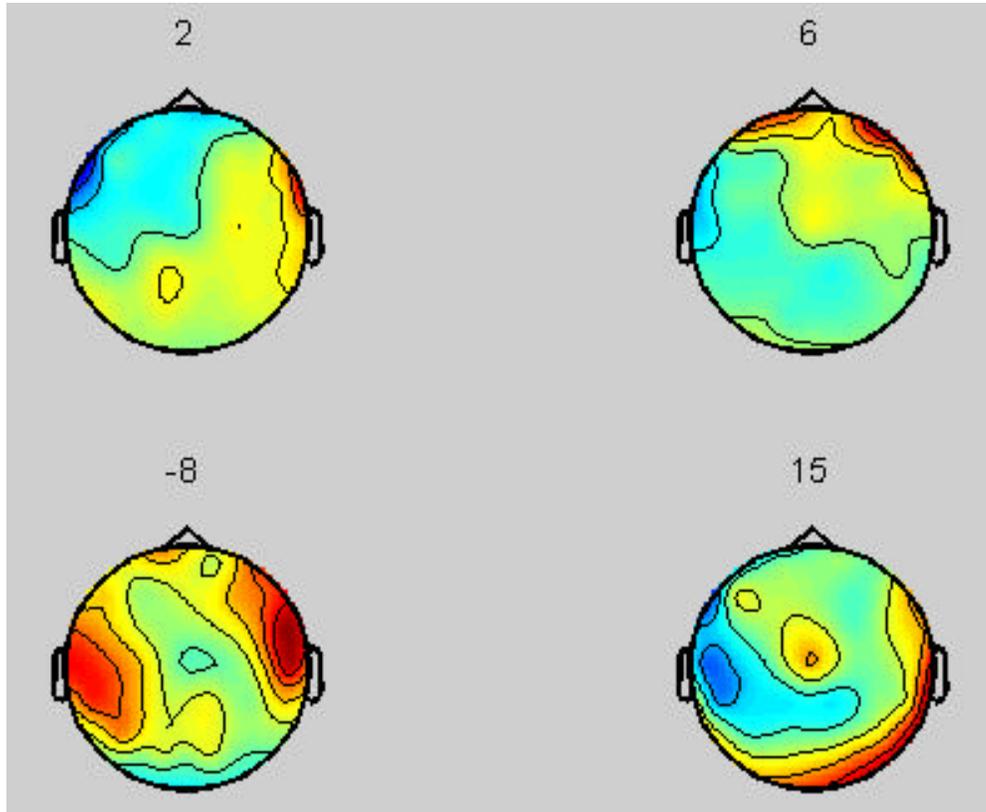


Figura 6-17 Mapas de algunas componentes

Se obtienen con claridad el HEOG, el ECG y la mordedura.

Al filtrar con HP 1 Hz se obtuvo:

- componente 2: VEOG
- componentes 1, 5, 6, 7: HEOG superpuesto a mordedura
- componente 15: HEOG superpuesto a otras componentes
- componente 3: ECG
- componente 4: mordedura
- componente 14: reloj (poco claro) y otras componentes
- resto: componentes no identificadas como artefactos

El VEOG se obtiene con claridad. El reloj no queda tan claro.

La ejecución duró 7 minutos.

6.2.14 fastica VIII

Las opciones utilizadas fueron:

- algoritmo fastica con deflación, 'gauss' y estabilización
- reducción a 15 componentes principales
- valor de stop 10^{-6}

Tras un tiempo de análisis de 7 minutos y 30 segundos los resultados fueron:

- componente 2: VEOG y ECG
- componente 10: VEOG y mordedura
- componente 6: HEOG
- componente 9: ECG

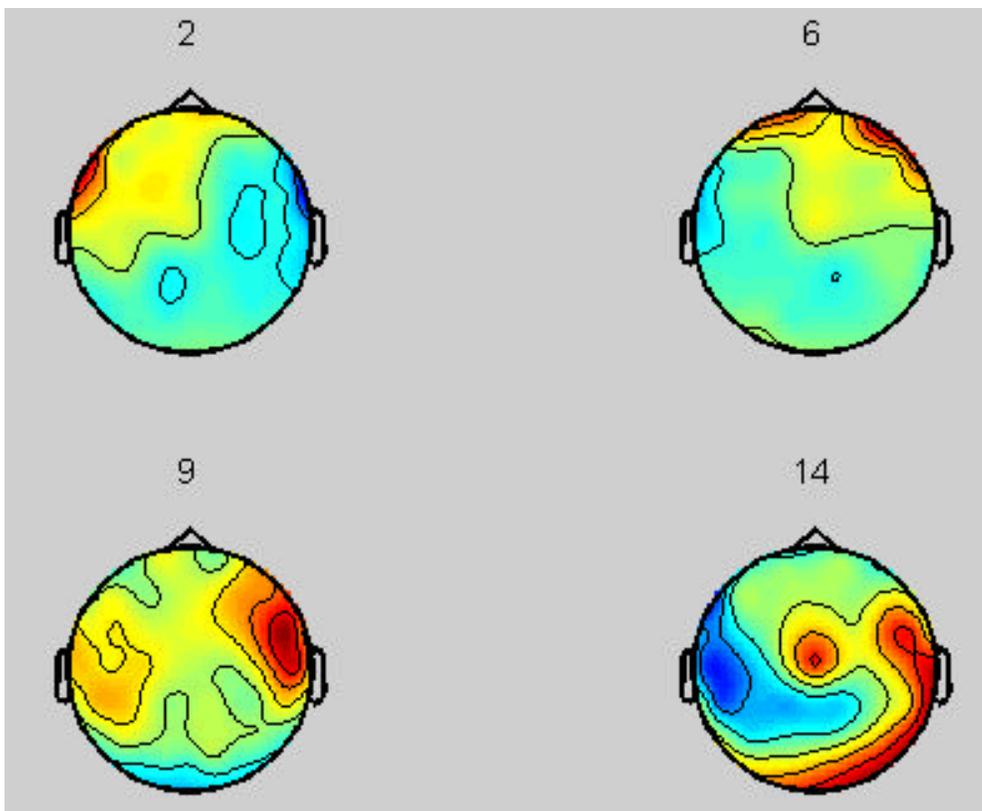


Figura 6-18 Mapas de algunas componentes obtenidas antes de filtrar

- componentes 1, 5, 7: mordedura
- componente 14: reloj (muy débil) y otras componentes

- resto: componentes no identificadas como artefactos.

Quedan claros el HEOG, el ECG y la mordedura.

Filtrando con HP 1 Hz se obtuvo:

- componente 2: VEOG
- componentes 1, 7: HEOG superpuesto a mordedura
- componente 8, 10, 11, 13, 15: HEOG y otras componentes
- componente 12: HEOG superpuesto a reloj y otras componentes
- componentes 4, 5: mordedura
- componentes 9: mordedura superpuesta a otras componentes
- resto: componentes no identificadas como artefactos.

El VEOG es obtenido con claridad. El reloj no es bien extraído.

El tiempo de ejecución fue de 6 minutos.

6.2.15 fastica IX

Las opciones utilizadas fueron:

- algoritmo fastica con deflación, 'skew' y estabilización
- reducción a 15 componentes principales
- valor de stop 10^{-6}

Tras un análisis de 7 minutos los resultados fueron:

- componente 2: VEOG y ECG (muy débil)
- componentes 5 y 8: HEOG superpuesto a respiración (5) y mordedura (8)
- componente 9: ECG
- componente 12: mordedura
- componentes 3, 6, 7: superposición de mordedura y respiración
- componentes 1, 4: superposición de mordedura y reloj
- componentes 13, 14, 15: superposición de mordedura, reloj y otras fuentes

- componente 10: respiración y otras componentes
- componente 11: no identificada como artefacto.

Se extrae con cierta claridad el VEOG y el HEOG. El ECG se extrae bien.

Al filtrar con HP 1 Hz se obtuvo:

- componente 2: VEOG
- componentes 5, 12, 14: superposición de HEOG y mordedura
- componentes 8, 10, 13, 15: superposición de HEOG, mordedura y reloj
- componentes 7, 11: superposición de HEOG, mordedura y otras componentes
- componente 3: mordedura
- componentes 6, 9: superposición de mordedura y otras componentes
- componente 4: no identificada como artefacto.

Se obtiene claramente el VEOG. El reloj con menos claridad.

El tiempo de análisis fue de 5 minutos.

6.2.16 extract_QN II

Las opciones utilizadas fueron:

- algoritmo extract_QN
- reducción a 15 componentes principales
- valor de stop 10^{-6}

Tras un análisis de 5 minutos los resultados obtenidos fueron:

- componente 3: VEOG, mordedura y ECG
- componente 1: HEOG
- componente 11: ECG
- componentes 4, 5, 6, 7, 8, 9 y 10: mordedura (aparece el reloj en la número 5 y la 10, respiración en la 7 y la 8, y otras componentes en la 9 y la 10)

- componentes 2, 14: superposición de reloj y otras componentes
- componente 12: respiración y otras componentes
- componentes 13, 15: no identificadas como artefactos.

Se obtiene una buena separación de HEOG, ECG y mordedura.

Al filtrar con HP 1 Hz se obtuvo, tras 4 minutos y 30 segundos:

- componente 2: VEOG
- componente 9: HEOG
- componentes 4, 5, 6, 7: superposición de HEOG y mordedura
- componente 3: ECG
- componentes 10, 12, 13, 15: superposición de mordedura y otras componentes

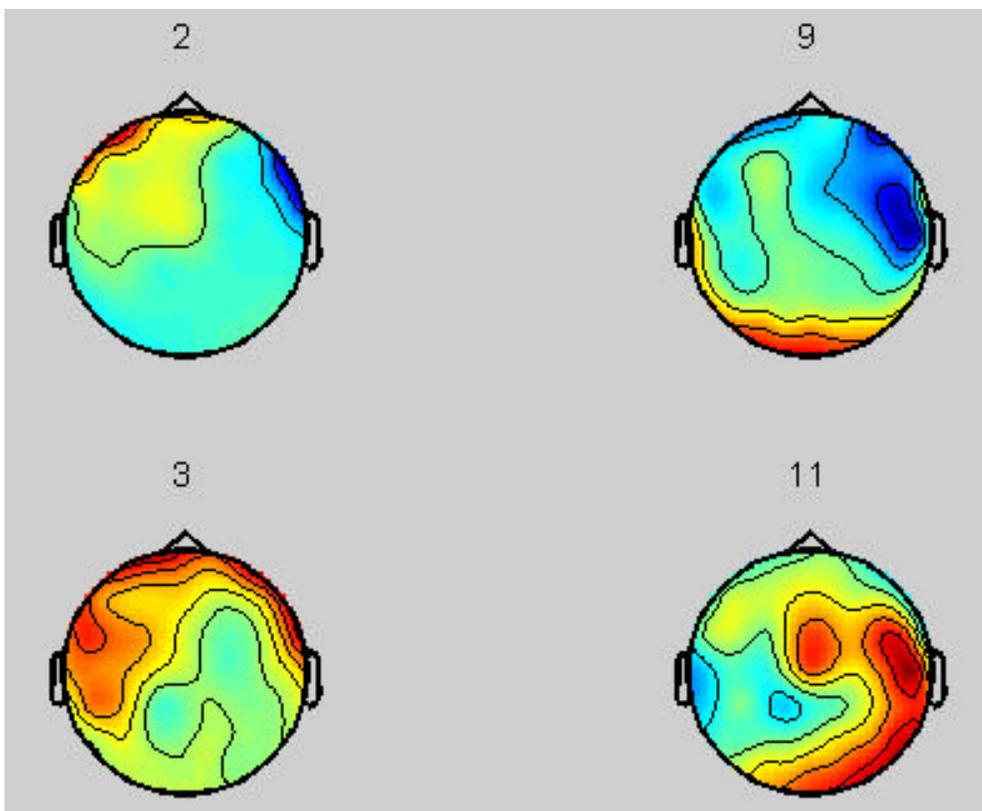


Figura 6-19 Mapas de algunas componentes obtenidas tras filtrar

- componente 1: mordedura
- componente 11: reloj y otras componentes

- componentes 8, 14: no identificadas como artefactos.

El VEOG se obtiene claramente. El reloj no demasiado.

6.2.17 extract2 II

Las opciones utilizadas fueron:

- algoritmo extract2
- reducción a 15 componentes principales
- valor de stop 10^{-6}

Tras un análisis de 3 minutos y 40 segundos los resultados fueron:

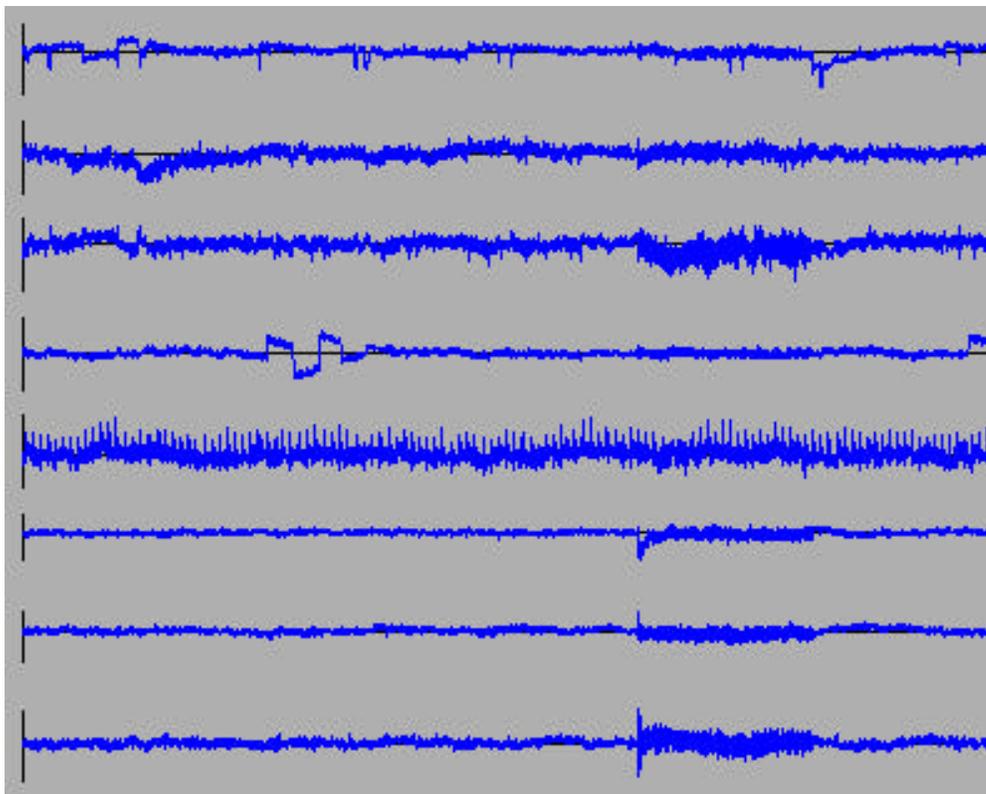


Figura 6-20 Componentes 2, 7, 8, 3, 6, 1, 9 y 4 obtenidas antes de filtrar

- componentes 2, 7 y 8: VEOG (ECG muy débil en la 2 y mordedura en 7 y 8)
- componente 3: HEOG

- componente 6: ECG
- componentes 1, 4, 5 y 9: mordedura (con reloj en la 4 y respiración en la 5)
- componentes 13, 15: superposición de mordedura y otras componentes
- componente 12: reloj y otras componentes.

Se obtiene el VEOG con bastante claridad. Se obtienen bien el HEOG, el ECG y la mordedura.

Al filtrar con HP 1 Hz , y tras 4 minutos de análisis, los resultados fueron:

- componente 3: VEOG
- componentes 4, 5 y 6: HEOG (con mordedura en la 4 y la 5 y reloj en la 6)
- componente 2: ECG
- componente 1: mordedura
- componentes 8, 9, 11, 12: superposición de mordedura y otras componentes
- componente 13: superposición de mordedura, reloj y otras componentes
- componentes 7, 10: superposición de reloj y otras componentes
- componentes 14, 15: no identificadas como artefactos.

Se obtiene bien el VEOG. El reloj no muy bien.

6.2.18 Comparación de las extracciones totales

El algoritmo más rápido es el INFOMAX sencillo de runica.

Sin filtrado HP, los tres casos en que se muestra más claro el VEOG son 6.2.1., 6.2.2. y 6.2.6. Los dos casos en que más claramente se aprecia el HEOG son 6.2.5. y 6.2.6., aunque los demás (en los que aparece HEOG) no difieren demasiado de estos dos. En los casos en que aparece HEOG, hay más de una componente en la que se encuentra.

El ECG es obtenido bastante bien por todos los algoritmos sin filtrado HP. También se obtiene bien la mordedura, aunque, a veces (como en 6.2.4., 6.2.6. y 6.2.7.), aparece también en otras salidas, impidiendo ver otras componentes con claridad.

Las respiraciones se separan muy bien, pero también a veces aparecen obstaculizando a otras componentes.

La componente más difícil de encontrar es, sin duda, el reloj digital. En los casos en que fue encontrada se identificó gracias al mapa cerebral, ya que la forma de onda no lo permitió.

En casi todos los casos las componentes más ruidosas (las no identificadas como artefactos) se encuentran entre las últimas extraídas (en el caso de las extracciones) o en las últimas filas de la matriz de separación (en el caso de las separaciones). El caso 6.2.3. es algo diferente, ya que hay muy pocas componentes no identificadas y demasiadas asociadas a respiración.

Las componentes más abundantes son las asociadas a respiraciones, y se sitúan más o menos en bloque por delante de las ruidosas.

Las componentes asociadas a la mordedura mejor separadas o extraídas se suelen encontrar entre las 40 primeras, exceptuando los casos 6.2.3. y 6.2.7., en que sólo una de estas componentes se halla entre las 40 primeras.

El ECG es encontrado entre las 11 primeras componentes en los algoritmos de runica, extract_QN y extract2, mientras que en los de fastica se encuentra entre la 47 (en 6.2.5.) y la 62 (en 6.2.4.).

Los algoritmos que encontraron VEOG y HEOG lo hicieron entre las primeras componentes (aunque el VEOG encontrado por 6.2.9. es la componente 66). Son llamativos los casos de 6.2.1. y 6.2.2., en que las 3 primeras componentes son VEOG (una de ellas) y HEOG (las otras dos), y el caso 6.2.9., en que las dos primeras componente son HEOG.

En todos los casos hubo que filtrar con filtro HP de 1 Hz para poder obtener con claridad el VEOG, el HEOG y la componente asociada al reloj digital.

Al filtrar los datos se obtienen resultados casi idénticos con todos los algoritmos, excepto la opción 'skew' (caso 6.2.7.) de fastica, cuyos resultados son peores: el HEOG aparece en dos componentes y de forma menos precisa; el reloj no tiene una forma de onda tan clara como en el resto de casos.

Exceptuando el caso 6.2.3. (fastica con separación, y, por lo tanto, no importa el orden) todos los algoritmos obtienen las componentes importantes (VEOG, HEOG y

reloj) entre las primeras. La mordedura y el ECG también se obtienen antes, pero, la calidad de las formas de onda de ambas (sobre todo del ECG) es mayor cuando no se filtra.

Por los buenos resultados y su altísima velocidad (en comparación con los demás algoritmos), el algoritmo INFOMAX sencillo de runica es el más recomendable.

También son recomendables, por ofrecer unos resultados igual de buenos, el ICA extended de runica, el extract_QN, el extract2 y la opción 'tanh' de fastica. Sin embargo los tiempos de ejecución son mucho más altos.

Merece la pena utilizar extract_QN o extract2 extrayendo solamente unas 35 componentes, reduciéndose así el tiempo de ejecución y obteniéndose las componentes de interés.

6.2.19 Comparación de las extracciones y separaciones con reducción de componentes principales

Cuando se realiza una reducción a 15 componentes principales, el VEOG es obtenido con bastante claridad en todos los algoritmos. Sin embargo, en el caso 6.2.10. tiene una fuerte componente de mordedura. En el resto tiene una componente del ECG, siendo los casos 6.2.12., 6.2.15. y 6.2.17. los que mejor extraen el VEOG. Además, excepto los casos 6.2.13., 6.2.15. y 6.2.16., aparece una componente VEOG en más de una salida, aunque sólo una es realmente clara.

El HEOG es obtenido muy bien por todos los algoritmos (el mejor es el de 6.2.13.), exceptuando el caso 6.2.15., en que aparece en dos salidas y con poca claridad.

El ECG es obtenido por todos los algoritmos de forma similar.

En todos los algoritmos hay al menos dos salidas cuyas formas de onda son las de las componentes asociadas a la mordedura. Sin embargo, estas componentes aparecen mezcladas con otras señales en otras salidas.

El reloj apenas es apreciable. Tan sólo en 6.2.12., 6.2.13. y 6.2.15. se puede observar alguna salida que tiene una forma de onda parecida a la del reloj, aunque en casi todos los casos es posible detectarlo en los mapas cerebrales.

Al filtrar con HP 1 Hz el VEOG es obtenido perfectamente por todos los algoritmos.

El HEOG es obtenido mejor en 6.2.10., 6.2.11., 6.2.16., 6.2.17. y 6.2.12 (en este orden). En el caso 6.2.15. no se tiene un buen resultado. En los casos 6.2.13. y 6.2.14. se obtienen salidas en que se superponen una componente con forma de HEOG bastante aceptable con otra componente (también fuerte) de mordedura. En todos los casos el HEOG aparece en más de una salida. De todos modos los resultados son peores que sin filtro.

El ECG obtenido es similar en todos los casos. Aparecen ondulaciones que empeoran la forma de onda con respecto al caso sin filtro. Sin embargo, los pulsos están mejor definidos.

En cuanto a las mordeduras, el comportamiento no es muy diferente al registrado en los casos sin filtro.

El reloj digital se obtiene con más claridad, pero el resultado nunca es tan aceptable como en los casos en que no se realizó reducción de componentes principales. Los peores casos son: 6.2.14. y 6.2.15. los mejores: 6.2.10. y 6.2.11.

Los tiempos de ejecución son pequeños (alrededor de los 5 minutos), exceptuando el caso 6.2.11. (el algoritmo ICA extendido de runica), que emplea media hora, con lo que la diferencia es enorme.

Los algoritmos más recomendables para realizar la reducción parecen ser el INFOMAX sencillo de runica y la opción 'tanh' de fastica. También se obtienen muy buenos resultados con extract_QN, extract2 y la opción 'gauss' de fastica. El algoritmo ICA extendido no es recomendable (a pesar de sus buenos resultados) por el elevado tiempo de ejecución.

La reducción de componentes principales ha demostrado ser muy útil, ya que, en poco tiempo y con pocas salidas, los algoritmos de ICA pueden obtener las fuentes originales con resultados aceptables, o incluso muy buenos, como en el caso del VEOG, en que los resultados con reducción y filtrado son tan buenos como sin reducción y con filtrado.

6.3 SEÑALES LIBRES DE ARTEFACTOS

Para eliminar los artefactos de los datos del MEG hay que hacer deflación de ellos. Es decir, hacer deflación con las filas de la matriz de separación correspondientes a los artefactos.

La matriz de separación es $\mathbf{W} = \mathbf{Q}\mathbf{B}$. En las simulaciones que hemos llevado a cabo, esta matriz \mathbf{W} es `weights*sphere`. Si llamamos \mathbf{W}_a a la matriz formada por las filas de \mathbf{W} correspondientes a los artefactos, la matriz de deflación \mathbf{P}_a será

$$\mathbf{P}_a = \mathbf{I} - \mathbf{W}_a^{\#}\mathbf{W}_a$$

En nuestro caso

$$\mathbf{P}_a = \text{eye}(\text{chans}) - \text{pinv}(\mathbf{W}_a) * \mathbf{W}_a$$

Los datos libres de artefactos, \mathbf{x}_L , serán

$$\mathbf{x}_L = \mathbf{P}_a \mathbf{x}$$

En nuestro caso

$$\text{dataL} = \mathbf{P}_a * \text{data}$$

De este modo se obtienen los datos sin las componentes asociadas a artefactos.

CAPÍTULO 7

CONCLUSIONES Y LÍNEAS FUTURAS

En este proyecto se ha realizado un estudio de la aplicación de la separación ciega de fuentes a señales de biomedicina, utilizando como ejemplo la aplicación de los distintos algoritmos estudiados a los datos procedentes de un MEG de 122 canales y 2 minutos de duración.

Han sido estudiados dos métodos diferentes de resolver el problema: la separación (BSS) y la extracción (BSE) con deflación.

El estudio ha sido realizado suponiendo mezclas lineales e instantáneas, ya que, en general (y en el caso concreto de un MEG), es una buena aproximación suponer que las señales biológicas recogidas por sensores en la superficie del cuerpo humano cumplen estas condiciones, debido a la alta velocidad de propagación (en comparación con las frecuencias máximas de esta señales).

Por lo tanto, la mezcla es la multiplicación de una matriz de mezclas constante por el vector de fuentes, y la separación es la multiplicación de una matriz de separación constante por el vector de datos (observaciones).

En estas condiciones la independencia 2 a 2 de las salidas da lugar a la separación de las fuentes en estas salidas, siempre que no exista más de una fuente Gaussiana. Para comprobar esta independencia es necesario utilizar estadísticos (momentos o cumulantes) de orden superior a 2 (HOS).

A pesar de no tenerse en cuenta el ruido aditivo, suponiendo que éste es Gaussiano, los métodos de separación son válidos en condiciones de ruido aditivo, por ser los HOS de señales Gaussianas nulos, lo que conduce a la separación de las fuentes no Gaussianas.

Aunque no se trata de una condición suficiente para la independencia 2 a 2, la decorrelación de los datos ha demostrado ser muy útil. Por una parte, facilita la elaboración de los algoritmos de separación y extracción. Por otra parte, utilizando una determinada manera de decorrelar (el método PCA, basado en la descomposición de la matriz de correlaciones de los datos en función de sus autovalores y autovectores), es posible retener una gran cantidad de información conservando únicamente las filas de la matriz de decorrelación correspondientes a los mayores autovalores (es decir, las componentes principales). Esto disminuye enormemente el tiempo de ejecución de la separación o de las extracciones y deflaciones, ya que es mucho menor el número de señales a las que se aplica el ICA.

En biomedicina es posible utilizar BSS o ICA con dos finalidades: la eliminación de artefactos y el estudio de las propias señales de interés.

El ejemplo realizado en este proyecto ha sido la eliminación de artefactos de los datos de un MEG. En este caso los artefactos suponían una gran mayoría de las fuentes, ya que se habían potenciado para estudiar la efectividad de los algoritmos en su separación.

En estudios del comportamiento cerebral los artefactos suponen un menor número de componentes, por lo que es posible realizar experimentos de respuestas a muchos estímulos simultáneos, separando mediante BSS o ICA las distintas respuestas.

Se han utilizado 4 algoritmos de separación y extracción. Tres de ellos (fastica, extract_QN y extract2) de extracción y el otro (runica) de separación.

El método runica utiliza el algoritmo INFOMAX, que desarrolla de dos maneras diferentes: el método del gradiente natural (que permite la separación de fuentes o bien subGaussianas o bien superGaussianas) y el método ICA extendido (que separa a la vez tanto fuentes subGaussianas como superGaussianas).

El programa fastica ofrece distintos algoritmos (incluyendo el FAST-ICA de Hyvärinen y Oja).

El toolbox ICA/EEG (concebido para su utilización en datos de EEG, pero también utilizable con datos de MEG) proporciona diversas funciones de representación de los datos y las salidas. De gran importancia es la representación de mapas cerebrales, que permiten localizar las distintas fuentes (componentes independientes). En el ejemplo desarrollado en el capítulo de simulaciones esto ha sido utilizado para

identificar algunas de las salidas cuyas formas eran dudosas. Este tipo de mapas es muy útil en el estudio del comportamiento cerebral (localización de la fuente de respuesta a un estímulo) o en la localización de lesiones o enfermedades.

En las simulaciones se ha podido comprobar cómo un número mayor de fuentes que de observaciones impide una correcta separación (ya que las fuentes de más no tienen porqué ser Gaussianas). El filtrado de las observaciones permite eliminar componentes y encontrar otras que estaban ocultas.

También se ha comprobado cómo un fenómeno fisiológico (como, por ejemplo, la mordedura o la respiración) da lugar a varias componentes independientes (la dependencia fisiológica no conlleva una dependencia temporal de las señales).

Se ha podido constatar la utilidad de los algoritmos de extracción diseñados para extraer en primer lugar fuentes de interés (con elevadas kurtosis), lo cual permite realizar menos extracciones (y hay que tener en cuenta que la tendencia es aumentar el número de sensores de los magnetómetros).

Quedan en biomedicina numerosas aplicaciones por estudiar. Para cada caso es necesario comprobar el cumplimiento de las hipótesis del ICA. Vimos que, por ejemplo, la detección y localización de tumores presentaba problemas. Ése es uno de los campos a investigar.

También hemos visto el caso de los spindles, útiles para la clasificación de las etapas del sueño. Aún no está resuelta la aplicación del ICA para el análisis de estos fenómenos.

Un problema que ha aparecido en las simulaciones de este proyecto es la identificación de algunas salidas, debido a que en ellas aparece más de una componente. La identificación es visual, por lo que sería útil una automatización de esa identificación. Ya se han realizado estudios al respecto, como en [Bauer2000].

APÉNDICE. NOTACIÓN Y ABREVIATURAS

NOTACIÓN

$()^\#$: pseudoinversa

A: matriz de mezcla

A': matriz de mezcla ampliada

A_E: matriz de proyección de E fuentes en las observaciones

B: matriz de preblanqueo

$C_{y_i y_j}^{\alpha, \beta}$: cumulante cruzado de orden $\alpha + \beta$

$C_{yy}^{\alpha\beta}$: matriz de cumulantes cruzados

$D(\parallel)$: divergencia de Kullback-Leibler

e: vector de ruido

$\phi_{\mathbf{v}}(\mathbf{w})$: primera función característica

G: matriz global de transferencia

G: matriz global de transferencia transparente

$h()$: entropía diferencial

I: matriz identidad

$I(Y_1, Y_2, \dots, Y_N)$: información mutua

K_s : kurtosis de s

$L(\mid)$: función de verosimilitud

$\mathcal{L}(\mid)$: función de verosimilitud logarítmica

λ : autovalor de la matriz de correlaciones

$M_{y_i, y_i}^{p, q}$: momento cruzado de orden $p + q$

P: matriz de deflación

p_s : f.d.p. conjunta de las fuentes

p_{s_i} : f.d.p. marginal de la fuente s_i

P_{s_i} : función de distribución (c.d.f.) de s_i

Q: matriz de separación aplicada a las observaciones decorreladas

Q_E : matriz de extracción de E fuentes aplicada a las observaciones decorreladas

R_{xx} : matriz de correlaciones de \mathbf{x}

\mathbf{s} : vector de fuentes

\mathbf{s}' : vector ampliado de fuentes

S_s : matriz diagonal de los signos de los cumulantes de \mathbf{s}

U: matriz de autovectores

V: matriz de mezcla de las fuentes en las observaciones decorreladas

V_E : matriz de proyección de E fuentes en las observaciones decorreladas

W: matriz de separación

W_E : matriz de extracción de E fuentes

\mathbf{x} : vector de observaciones

\mathbf{x}' : vector de observaciones decorreladas

\mathbf{x}_D : vector de observaciones tras deflación

\mathbf{y} : vector de salidas

$\psi_v(\mathbf{w})$: segunda función característica

$\Psi(\mathbf{Y})$: función contraste

ABREVIATURAS

- ABP: Arterial Blood Pressure (presión arterial de la sangre)
- ANS: Autonomic Nervous System (sistema nervioso autónomo)
- BOLD: Blood Oxygenation Level Dependent (contraste dependiente del nivel de la oxigenación de la sangre)
- BSE: Blind Source Extraction (Extracción Ciega de Fuentes)
- BSS: Blind Source Separation (Separación Ciega de Fuentes)
- BSSS: Blind Source Subspace Separation (Separación Ciega de Subespacios de fuentes)
- CII: Cumulant-based Iterative Inversion (inversión iterativa basada en cumulantes)
- CPNA: Cardiac Sympathetic Nervous Activity (actividad nerviosa simpática cardiaca)
- CSNA: Cardiac Parasympathetic Nervous Activity (actividad nerviosa parasimpática cardiaca)
- ECG: Electrocardiograma
- EEG: Electroencefalograma
- ENTMAX: Entropy Maximization (maximización de entropía)
- EOG: Electrooculograma
- ERP: Event-Related Potential (potencial asociado a evento)
- f.d.p.: función de densidad de probabilidad
- FECG: Fetal ECG (ECG del feto)
- FHR: Fetal Heart Rate (frecuencia cardiaca del feto)
- fMRI: functional MRI (MRI funcional)
- HEOG: Horizontal EOG (EOG horizontal)
- HOS: Higher Order Statistics (estadísticos de orden superior)

ICA: Independent Component Analysis (Análisis de Componentes Independientes)

INFOMAX: Information Maximization (maximización de la información)

LTI: Lineal e invariante en el tiempo

ME: Maximización de la entropía

MECG: Maternal ECG (ECG de la madre)

MEG: Magnetoencefalograma

MI: Mutual Information (información mutua)

MICA: Multidimensional ICA (ICA multidimensional)

ML: Maximum Likelihood

MMI: Minimum Mutual Information (mínima información mutua)

MRI: Magnetic Resonance Imaging (obtención de imágenes por resonancia magnética)

PCA: Principal Component Analysis (Análisis de Componentes Principales)

ROI: Region of Interest (región de interés)

SQUID: Superconducting Quantum Interference Device

VEOG: Vertical EOG (EOG vertical)

REFERENCIAS

- [1] [<http://www.cnl.salk.edu/~scott/ica.html>] Scott Makeig, Computational Neurobiology Lab, Salk Institute, La Jolla CA, USA
- [2] [<http://www.cis.hut.fi/rvigario>] Ricardo Vigário, Laboratory of Computer and Information Science, Helsinki University of Technology, Finland
- [3] [lauri@neuro.hut.fi] Lauri Parkkonen, Brain Research Unit, Helsinki University of Technology, Finland
- [Barros2000] Barros, A. et al. (2000) Extraction of Event-Related Signals from multichannel Bioelectrical Measurements, *IEEE transactions on biomedical engineering*, 47 (5): 583-588.
- [Bauer2000] Bauer, Ch. et al. (2000) Can ICA help identify brain tumor related EEG signals?, *Proceedings of the Second International Workshop on ICA and BSS, ICA' 2000*, Helsinki, Finland, June 2000, pp. 609-614.
- [Bell95] Bell, A. y Sejnowski, T. (1995) An Information-Maximization Approach to Blind Separation and Blind Deconvolution, *Neural Computation* 7: 1129-1159.
- [Bell2000] Bell, A.(2000) Information Theory, Independent-Component Analysis and applications. En *Unsupervised adaptive filtering*, vol. I, pp. 237-264. Ed. S. Haykin, Willey & Sons.

REFERENCIAS

- [Cruces1999] Cruces, S. (1999) *Una visión unificada de los algoritmos de separación ciega de fuentes*, Tesis Doctoral, Universidad de Vigo, Departamento de Procesado de señal.
- [Cruces2000] Cruces, S., Cichcki, A. y Castedo, L. (2000) Blind Source Extraction in Gaussian noise, *Proceedings of the second International Workshop on ICA and BSS, ICA' 2000*, Helsinki, Finland, June 2000, pp. 63-68.
- [Cruces2001] Cruces, S., Cichocki, A., Amari, S. y Castedo, L. (2001) Simultaneous blind extraction of arbitrary group of sparse signals, *RIKEN* Technical report.
- [DeLathauwer2000] De Lathauwer, L., De Moor, B. y Vandewalle, J. (2000) Fetal electrocardiogram extraction by Blind Source Subspace Separation, *IEEE transactions on biomedical engineering*, 47 (5): 567-572.
- [Hari97] Hari, R. and Salmelin, R. (1997) Human cortical oscillations: a neuromagnetic view through the skull, *TINS*, 20 (1): 44-49.
- [Lee98] Lee, T.-W. et al. (1998) Biomedical applications of ICA, *Independent Component Analysis*, pp. 145-146. Ed. Kluwer Academic Publishers.
- [Jung2000] Jung, T.-P. et al. (2000) Independent Component Analysis of biomedical signals, *Proceedings of the second International Workshop on ICA and BSS, ICA' 2000*, Helsinki, Finland, June 2000, pp. 633-644.
- [Rosipal98] Rosipal, R. et al. (1998) Can ICA improve sleep-spindles detection?, *Neural Network World* 5: 539-547.
- [Vigário97] Vigário, R. et al. (1997) Independent Component Analysis for identification of artifacts in magnetoencephalographic recordings, *Neural Information Processing Systems 10, Proceedings of NIPS' 97*, Denver, December 1997.

- [Vigário2000] Vigário, R. et al. (2000) Independent Component Analysis Approach to the Analysis of EEG and MEG Recordings, *IEEE transactions on biomedical engineering*, 47 (5): 589-593.
- [Virag2000] Virag, N. et al. (2000) Observer of an autonomic cardiac outflow based on Blind Source Separation of ECG parameters, *IEEE transactions on biomedical engineering*, 47 (5): 578-582.