3. Obtención de las componentes independientes en ICA

3.1 Introducción

En esta sección veremos algunos de los diferentes métodos y algoritmos existentes que permiten obtener la matriz de la transformación *ICA*. Previamente veremos algunos aspectos que pueden resultar de interés de cara a una correcta aplicación de dichas técnicas.

Para desarrollar este apartado, nos basaremos esencialmente en [Hyvärinen01] y [Jenssen00], aunque en algunos momentos sea necesario recurrir a otros libros o artículos para destacar ciertos aspectos concretos.

3.2 Preprocesado

En este apartado vamos a ver algunas operaciones previas que se tendrán que aplicar a los datos observados para que los algoritmos que veremos posteriormente funcionen de forma adecuada.

De acuerdo a las ambigüedades expuestas en la sección anterior referentes al análisis *ICA*, el hecho de fijar la varianza de las fuentes originales s_i a la unidad se transformaba en una ambigüedad de signo en las componentes independientes estimadas y_i, es decir, que tendremos unas estimaciones $y_i = \pm s_i$. Además, si las componentes s_i eran independientes, esto implicaba que también eran incorreladas. El modelo *ICA*, atendiendo a que supondremos que los datos $\boldsymbol{x}$ que le presentamos están blanqueados y con varianza fijada a la unidad y que para no confundirnos a partir de ahora denotaremos por $\boldsymbol{z}$, se puede expresar del siguiente modo, con una matriz de transformación ortonormal **W'** (con el fin de conseguir que las componentes s_i sean independientes, incorreladas y con varianza unidad):

$$\mathbf{s} = \mathbf{W}'\mathbf{z} \tag{3.1}$$

Los datos blanqueados $\boldsymbol{z}$, que además están centrados, (es decir, que forzamos a que tengan media cero), son los que realmente le presentamos al método ICA para estimar **W'**, de forma que $\boldsymbol{y} = \mathbf{W'}\ \boldsymbol{z}$, con $\boldsymbol{y}$ componentes lo más independientes entre sí que sea posible lograr. Dado que la matriz **W'** sólo trabaja con datos blanqueados, le aplicaremos el mismo método que le aplicamos a $\boldsymbol{x}$ (datos originales) para blanquearlos en $\boldsymbol{z}$ y así obtener la matriz real de la transformación, **W**, que nos llevará realmente $\boldsymbol{x}$ al nuevo espacio ICA, de acuerdo a $\boldsymbol{y} = \mathbf{W}\,\boldsymbol{x}$ [Jenssen00].

3.2.1 Centrado

Sin pérdida de generalidad, podemos considerar que las variables aleatorias resultado de la mezcla y las componentes independientes tienen media cero, lo cual permite simplificar la teoría y los algoritmos [Hyvärinen01].

Si esta hipótesis no es cierta, es posible realizar un preprocesado previo para hacer que se cumpla. Esto se consigue *centrando* las variables observadas, esto es, quitándole la media. Esto significa que las observaciones originales (denotémoslas por ***x'***), quedarán de la siguiente manera:

$$\mathbf{x} = \mathbf{x}' - E\{\mathbf{x}'\} \tag{3.2}$$

De esta forma, las componentes independientes tendrán también media cero debido a que:

$$E\{\mathbf{s}\} = \mathbf{A}^{-1}E\{\mathbf{x}\} \tag{3.3}$$

Por otro lado, la matriz de mezcla permanecerá inalterada después del proceso de centrado, por lo que este paso siempre se podrá realizar sin temor alguno de afectar a dicha matriz. Tras estimar la matriz de mezcla y las componentes independientes a partir de los datos centrados, la media eliminada se puede añadir con tan sólo sumar $\mathbf{A}^{-1}E\{\mathbf{x}\}$ a las variables estimadas (que tendrán media cero).

3.2.2 Blanqueo

El proceso de blanqueo permite que los datos resulten incorrelados entre si y que además tengan media cero, todo ello tras aplicarle una transformación que denotaremos por **V**, como vimos en el apartado correspondiente relativo a *PCA*:

$$\mathbf{z} = \mathbf{V}\mathbf{x} \tag{3.4}$$

de forma que se verificará que:

$$E\{\mathbf{z}\mathbf{z}^{\mathrm{T}}\} = \mathbf{I} \tag{3.5}$$

siendo **I** la matriz identidad y **V** la transformación que permite blanquear los datos de ***x*** en ***z***.

Además del *blanqueo PCA* que ya fue estudiado anteriormente, existe una nueva técnica que permite obtener idénticos resultados y es la que se conoce como *blanqueo simétrico,* que estudiaremos seguidamente.

3.2.2.1 Blanqueo simétrico

Para obtener la matriz de transformación **V** se seguirá el siguiente razonamiento basado en la obtención de autovalores y autovectores:

$$\mathbf{V} = E\{\mathbf{x}\mathbf{x}^T\}^{-\frac{1}{2}} = \mathbf{R}_{\mathbf{x}}^{-\frac{1}{2}} = (\mathbf{U}\mathbf{D}\mathbf{U}^T)^{-\frac{1}{2}} = \mathbf{U}\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T \tag{3.6}$$

siendo $\mathbf{R_x}$ la matriz de covarianzas de los datos sin blanquear y **D** y **U**, como es sabido, las matrices que contienen los autovalores y autovectores respectivamente.

3.2.2.2 El blanqueo es sólo parte de *ICA*

A continuación, supongamos que los datos del modelo han sido blanqueados por cualquiera de los métodos anteriores. Por tanto, a partir de las ecuaciones que se acaban de exponer se puede ver como:

$$\mathbf{z} = \mathbf{VAs} = \mathbf{\Delta s} \tag{3.7}$$

donde $\mathbf{\Delta}$ es la nueva matriz de mezcla resultado del proceso de blanqueo. Podríamos pensar que el blanqueo resuelve el problema *ICA* debido a que la incorrelación está relacionada con la independencia. Sin embargo esto no es del todo cierto. La incorrelación es una condición más débil que la independencia y por si sola no es suficiente para estimar la matriz de separación. Para verlo, consideremos una transformación *ortogonal* **P** de **z**:

$$\mathbf{y} = \mathbf{Pz} \tag{3.8}$$

y debido a la ortogonalidad de **P**, tendremos que:

$$E\left\{\mathbf{yy}^T\right\} = E\left\{\mathbf{Pyy}^T\mathbf{P}^T\right\} = \mathbf{PIP}^T = \mathbf{I} \tag{3.9}$$

Es decir, ***y*** está también blanqueada. Por tanto no estaremos en disposición de ver si las componentes independientes vienen dadas por ***z*** o bien por ***y*** usando tan sólo la propiedad de blanqueo. Además, como ***y*** se puede obtener a partir de cualquier transformación ortogonal de ***z***, entonces podemos afirmar que *el blanqueo proporciona las componentes independientes tan sólo a partir de una transformación ortogonal*. Sin embargo, esto no es suficiente en la mayoría de las aplicaciones.

Por otro lado, el blanqueo es de enorme utilidad como un paso previo antes de comenzar con la resolución del modelo *ICA*. La utilidad del blanqueo consiste en que la nueva matriz de mezcla $\mathbf{\Delta} = \mathbf{VA}$ es *ortogonal*. Esto se puede demostrar a partir del siguiente razonamiento:

$$E\left\{\mathbf{zz}^T\right\} = \mathbf{\Delta}E\left\{\mathbf{zz}^T\right\}\mathbf{\Delta}^T = \mathbf{\Delta\Delta}^T = \mathbf{I} \tag{3.10}$$

Esto implica que podemos restringir nuestra búsqueda de la matriz de mezcla al espacio de las matrices ortogonales. Es decir, en vez de tener que buscar los n^2 que contendría la matriz original, ahora tan sólo tendremos que movernos en los $\frac{n(n-1)}{2}$ grados de libertad posibles. Por ejemplo, para dos dimensiones, habrá que calcular un solo elemento, mientras que para dimensiones mayores, la matriz ortogonal tendrá aproximadamente la mitad de elementos que los que tendría una matriz cualquiera que no lo fuera.

De esta manera podremos afirmar que el blanqueo sólo va a resolver la mitad del problema *ICA* ya que tan sólo nos va a ayudar a reducir el número de parámetros a encontrar a la mitad, los cuales habrán de ser estimados por diferentes algoritmos que estudiaremos posteriormente [Hyvärinen01].

3.2.2.3 ¿Por qué las variables gaussianas no son válidas en *ICA*?

Para verlo, consideremos que la densidad de probabilidad conjunta de dos variables aleatorias independientes s_1 y s_2, es gaussiana. Esto significa que dicha densidad de probabilidad resultará:

$$p(s_1, s_2) = \frac{1}{2\pi}\exp(-\frac{{s_1}^2 + {s_2}^2}{2}) = \frac{1}{2\pi}\exp(-\frac{\|s\|^2}{2}) \tag{3.11}$$

A continuación, asumamos que la matriz de mezcla **A** es ortogonal. Por ejemplo, podríamos considerar que es debido a que los datos han sido blanqueados. Usando expresiones de estadística básica y observando que por ser ortogonal, se cumple que $\mathbf{A}^{-1} = \mathbf{A}^T$, llegaremos a que la densidad de probabilidad conjunta de las señales mezcladas $\boldsymbol{x_1}$ y $\boldsymbol{x_2}$ resultará:

$$p(x_1, x_2) = \frac{1}{2\pi}\exp(-\frac{\|\mathbf{A}^T\mathbf{x}\|^2}{2})\left|\det \mathbf{A}^T\right| \tag{3.12}$$

Debido a la ortogonalidad de **A**, tendremos que $\|\mathbf{A}^T\mathbf{x}\|^2 = \|\mathbf{x}\|^2$ y que $\left|\det \mathbf{A}\right| = 1$ y sabiendo además que si **A** ortogonal también lo será $\mathbf{A}^T$, tendremos que:

$$p(x_1, x_2) = \frac{1}{2\pi}\exp(-\frac{\|\mathbf{x}\|^2}{2}) \tag{3.13}$$

y comprobaremos como la matriz de mezcla **A** no aparece en la densidad de probabilidad conjunta, por lo que a partir de (3.13) y (3.11), vemos que las distribuciones original y de las señales mezcladas son idénticas. Sin embargo no existe ningún método que permita obtener la matriz a partir de las mezclas [Hyvärinen01].

El fenómeno que explica que las matrices de mezcla ortogonales no se pueden calcular a partir de variables gaussianas, está relacionado con la propiedad que indica que variables gaussianas incorreladas entre si son necesariamente independientes. Por tanto, la información que reside en la independencia no nos permite llegar más allá del blanqueo. En otras palabras, la matriz de mezcla **A** no se puede calcular para variables gaussianas independientes. Con las variables gaussianas a lo más que podemos llegar es a blanquear los datos, por ejemplo usando *PCA*.

¿Qué ocurre si intentamos estimar el modelo *ICA* cuando algunas de las componentes son gaussianas y otras no gaussianas? En este caso, podremos estimar las componentes no gaussianas y las gaussianas no se podrán separar entre ellas. Es decir, algunas de las componentes estimadas, serán combinaciones lineales de componentes

gaussianas. En realidad, esto implica que en el caso de una sola variable gaussiana, podremos estimar el modelo ya que no existe ninguna otra componente gaussiana que se pudiera mezclar con ella.

3.2.3 Inversión del blanqueo en W'

Una vez llevemos a cabo la estimación de **W'** por el algoritmo ICA en cuestión, tendremos que realizar la trasformación inversa a ella con el fin de poder llevar los datos originales de media cero ***x*** al espacio ICA. Suponiendo que el blanqueo lo hemos llevado a cabo por el método PCA, será necesario realizar el siguiente procedimiento:

$$\mathbf{y} = \mathbf{W}'\mathbf{z} = \mathbf{W}'\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{x} = \mathbf{W}\mathbf{x} \tag{3.14}$$

de forma que por comparación, llegaremos a que la matriz de estimación *ICA* **W**, será:

$$\mathbf{W} = \mathbf{W}'\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T \tag{3.15}$$

y será la última operación que tendremos que llevar a cabo en el método *ICA* [Jenssen00].

3.3 Estimación del modelo *ICA* a partir de la maximización de la no gaussianidad

En este apartado vamos a ver un principio muy simple e intuitivo que nos va a permitir estimar el modelo del *Análisis de Componentes Independientes (ICA)* y que estará basado en la maximización de la no gaussianidad. La no gaussianidad es un parámetro de vital importancia en la estimación *ICA* y sin el cual ésta no sería posible.

3.3.1 La no gaussianidad conlleva independencia

Como ya hemos visto previamente, el *Teorema Central del Límite* afirma que una distribución que es suma de variables aleatorias tiende a una distribución gaussiana bajo ciertas condiciones. Consideremos que el vector de observaciones ***x*** se genera a partir del modelo *ICA* como:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{3.16}$$

Asumamos a su vez que todas las componentes independientes a estimar presentan la misma distribución. Por tanto, la estima de dichas componentes vendría dada por la obtención de la inversa de la matriz de mezcla:

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{x} \tag{3.17}$$

Para estimar una de las componentes independientes podríamos considerar una combinación lineal de las x_i. Llamemos $y = \mathbf{b}^T\mathbf{x} = \mathbf{b}^T\mathbf{A}\mathbf{s} = \sum_i b_i x_i$, donde **b** es un vector a determinar. Por tanto, y será una combinación lineal de las x_i, donde los coeficientes de la mezcla vienen dados por $\mathbf{b}^T\mathbf{A}$, que llamaremos de aquí en adelante $\mathbf{q}^T$. De esta forma tendremos:

$$y = \mathbf{b}^T\mathbf{x} = \mathbf{q}^T\mathbf{s} = \sum_i q_i s_i \tag{3.18}$$

Si **b** fuera una de las filas de la inversa de **A**, entonces la combinación lineal $\mathbf{b}^T\mathbf{x}$ coincidiría con una de las componentes independientes. En este caso, el correspondiente vector $\mathbf{q}^T$ tendría todos sus elementos a cero salvo uno de ellos que valdría '1'.

¿Cómo hemos de usar el Teorema Central del Límite para determinar **b** de forma que coincida con una de las filas de la inversa de **A**? En realidad no es posible hacerlo directamente porque en principio no sabemos nada de la matriz de mezcla, pero sería posible obtener un *estimador* que nos proporcionara una buena aproximación [Hyvärinen01].

Variemos los coeficientes del vector **q** y veamos como cambia $y = \mathbf{q}^T\mathbf{s}$. La idea consiste en ver que dado que la suma de dos variables aleatorias es más gaussiana que las variables originales, entonces $y = \mathbf{q}^T\mathbf{s}$ será más gaussiana que las s_i, siendo lo menos gaussiana posible cuando y coincide con una de las s_i. En este caso, evidentemente, sólo uno de los elementos de **q** es distinto de cero.

No sabemos en la práctica los valores de **q**, pero esto no tiene demasiada importancia ya que $\mathbf{q}^T\mathbf{s} = \mathbf{b}^T\mathbf{x}$ por la propia definición de **q**. Por tanto, podremos variar **b** y ver como lo hace a su vez $\mathbf{b}^T\mathbf{x}$.

Sin embargo, podemos tomar **b** como un vector que maximiza la no gaussianidad de $\mathbf{b}^T\mathbf{x}$ y que necesariamente se tendrá que obtener a partir de $\mathbf{q} = \mathbf{A}^T\mathbf{b}$, que sólo tiene una componente distinta de cero. Esto significa que $y = \mathbf{b}^T\mathbf{x} = \mathbf{q}^T\mathbf{s}$ se corresponde con una de las componentes independientes. Es decir, maximizando la no gaussianidad de $\mathbf{b}^T\mathbf{x}$ podemos estimar una de estas componentes.

En realidad, la optimización para maximizar la no gaussianidad en el espacio de dimensión n de vectores **b** tiene $2n$ máximos, dos por cada componente independiente y que se corresponden con s_i y con $-s_i$ (ya que como vimos, las componentes independientes se pueden estimar de forma cerrada salvo un signo multiplicativo).

El principio de máxima no gaussianidad se puede explicar a partir de algunos ejemplos que veremos seguidamente.

Ejemplo

Sean dos componentes independientes con distribución uniforme de media cero. La distribución conjunta se ilustra en la siguiente figura donde se ha colocado en el eje horizontal la primera componente independiente x_1 y en el vertical la segunda, x_2.

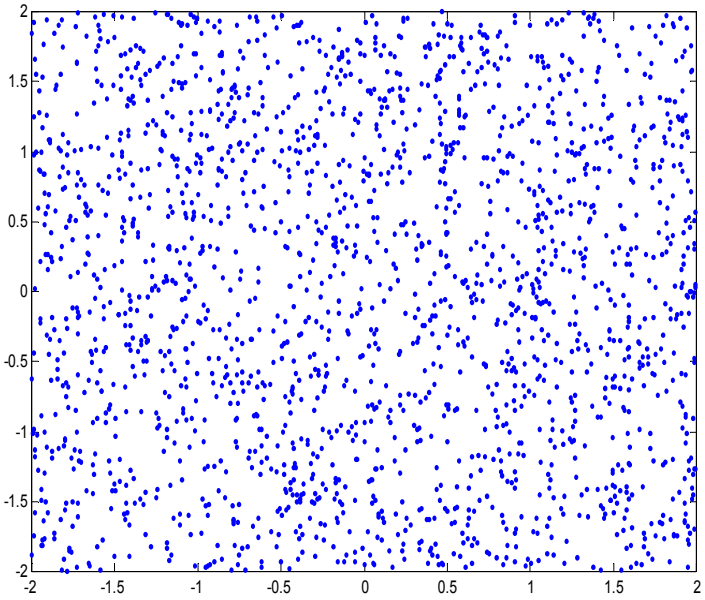

Figura 3.1 *Distribución conjunta de dos componentes independientes con densidad uniforme. En el eje horizontal está la primera componente independiente x_1 y en el vertical la segunda, x_2. Se suponen uniformes en el intervalo [-2,2].*

A continuación vamos a comprobar como la nueva densidad resultante de sumar las dos distribuciones anteriores se parece más a una gaussiana que la original.

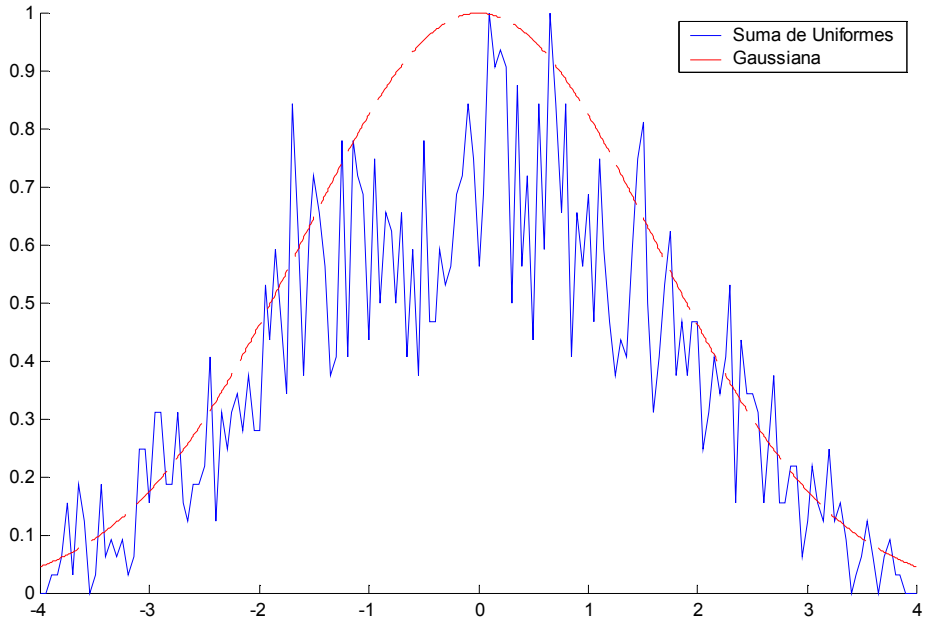

Figura 3.2 *Comprobación del Teorema Central del Límite. En esta figura vemos en trazo azul el resultado de sumar las dos variables uniformes anteriores y en trazo rojo discontinuo la densidad gaussiana equivalente que presenta la misma media y varianza que la suma de las uniformes. Podemos ver como la suma de uniformes tiende a parecerse más a la gaussiana.*

Si ahora realizamos el proceso de blanqueo, vamos a verificar como las densidades marginales también se parecen más a una gaussiana que antes. Para ello vamos a realizar un blanqueo *PCA* siguiendo los pasos que se estudiaron en el apartado correspondiente.

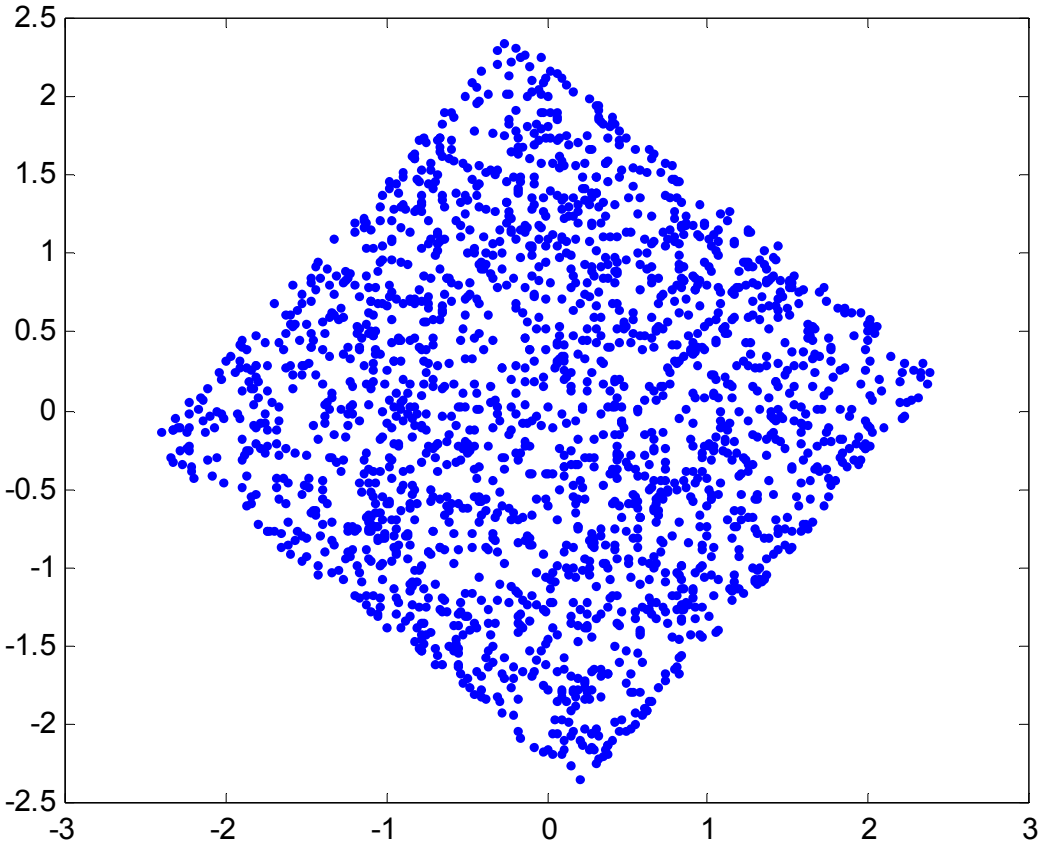

Figura 3.3 *Distribución conjunta de los datos tras ser blanqueados. En el eje horizontal aparece* z_1 *y en el vertical* z_2.

Por último, comprobaremos que las densidades marginales tras el proceso de blanqueo son más parecidas a una densidad gaussiana que las densidades correspondientes a las componentes independientes:

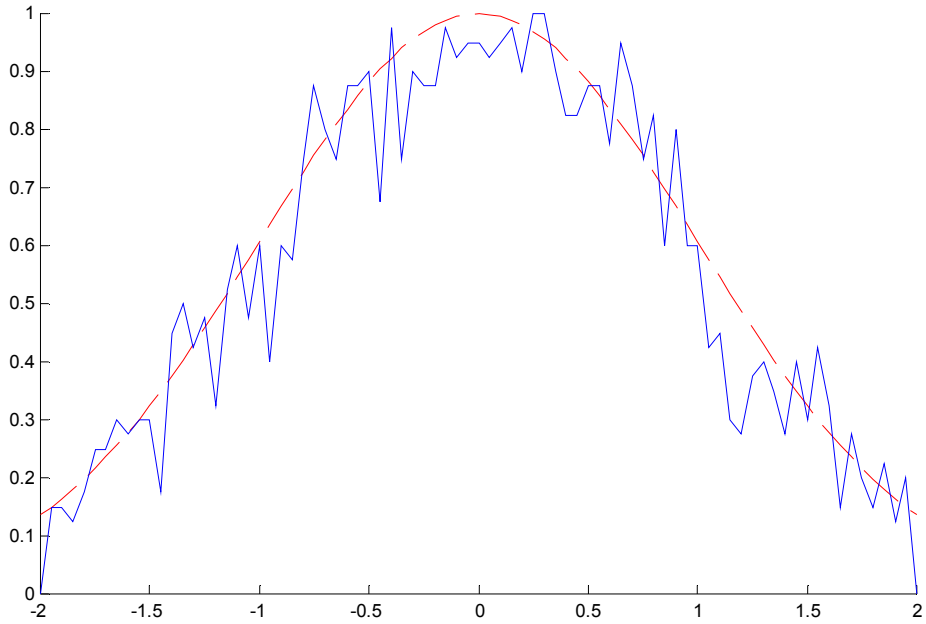

Figura 3.4 *Tras el proceso de blanqueo las densidades marginales resultan más parecidas a la densidad gaussiana.*

3.3.2 El método del gradiente y la kurtosis

3.3.2.1 Los máximos y mínimos de la kurtosis dan lugar a las componentes independientes

Como ya vimos en apartados previos referentes a la kurtosis, ésta permite establecer una medida de la gaussianidad de una cierta variable aleatoria. Seguidamente vamos a ver como es posible estimar las componentes independientes a partir de la minimización o maximización de la kurtosis. Para desarrollar el razonamiento y fijar ideas, procederemos con un modelo *ICA* de dos dimensiones del tipo $\mathbf{x} = \mathbf{A}\mathbf{s}$. Consideremos también que las dos componentes independientes s_1 y s_2 tienen kurtosis que denotaremos por $kurt(s_1)$ y $kurt(s_2)$ respectivamente y que serán distintas de cero. Remarquemos además que tienen varianza unidad. En este problema tratamos de encontrar una de las dos componentes independientes de la forma $y = \mathbf{b}^T\mathbf{x}$.

Además por ser $\mathbf{q} = \mathbf{A}^T\mathbf{b}$, tendremos que $y = \mathbf{b}^T\mathbf{x} = \mathbf{b}^T\mathbf{A}\mathbf{s} = \mathbf{q}^T\mathbf{s} = q_1 s_1 + q_2 s_2$. Y basándonos en la propiedad aditiva de la kurtosis [Hyvärinen01], podremos llegar al siguiente resultado:

$$kurt(y) = kurt(q_1 s_1) + kurt(q_2 s_2) = q_1^4 kurt(s_1) + q_2^4 kurt(s_2) \tag{3.19}$$

Por otro lado, podemos decir que la varianza de y es la unidad si hemos normalizado lo cual nos lleva a una nueva restricción en y:

$$E\{y^2\} = q_1^2 + q_2^2 \tag{3.20}$$

que desde el punto de vista geométrico, nos lleva a pensar que **q** se encuentra restringida al círculo unidad en el plano 2D.

El problema de optimización consistirá ahora en encontrar el máximo de $\left|kurt(y)\right| = \left|q_1^4 kurt(s_1) + q_2^4 kurt(s_2)\right|$ en el círculo unidad. Para empezar, podemos considerar que las kurtosis valen uno, por lo que la nueva función a maximizar quedará como:

$$F(\mathbf{q}) = q_1^4 + q_2^4 \tag{3.21}$$

En la siguiente figura, podemos ver algunas trazas de esta función de dos variables:

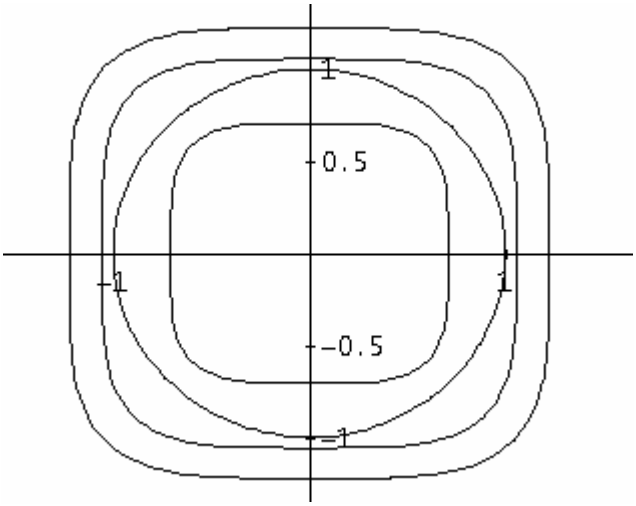

Figura 3.5 *Algunas trazas de la función F(**q**) representadas junto al círculo unidad.*

A partir de la figura anterior, es fácil deducir que los máximos se van a dar cuando uno de los elementos de **q** es cero y el otro es '1' o '-1', debido a la restricción que impone el círculo unidad. Pero esos puntos además serán aquellos en los que *y* coincide con una de las componentes independientes $\pm s_i$, por lo que el problema estaría resuelto.

El hecho de que las kurtosis tomaran un valor '-1' no cambia nada la situación ya que al tomar valores absolutos, obtendríamos la misma función a maximizar. Por último, si las kurtosis tomaran valores completamente arbitrarios (pero nunca cero), se puede demostrar que el máximo seguiría estando cuando $y = \mathbf{b}^T \mathbf{x}$.

Veamos seguidamente la utilidad que tiene el *blanqueo*. Para los datos blanqueados **z**, tratamos de buscar una combinación lineal $\mathbf{w}^T \mathbf{z}$ (siendo **w** el vector que permite extraer la componente independiente), que maximice la no gaussianidad. En nuestro caso, veremos como la situación se simplifica ya que tenemos que $\mathbf{q} = (\mathbf{VA})^T$ y por tanto:

$$\left\|\mathbf{q}\right\|^2 = (\mathbf{w}^T \mathbf{VA})(\mathbf{A}^T \mathbf{V}^T \mathbf{w}) = \left\|\mathbf{w}\right\|^2 \tag{3.22}$$

lo cual nos lleva a que la restricción de que **q** ha de estar contenido en la esfera unidad, coincide con que **w** ha de estar también sobre la esfera de radio unidad.

Es decir, que simplemente tendremos que maximizar el valor absoluto de la kurtosis de $\mathbf{w}^T\mathbf{z}$ sujeto a la restricción de que $\|\mathbf{w}\|=1$. Además después del blanqueo veremos que el término $\mathbf{w}^T\mathbf{z}$ se puede interpretar como la proyección sobre la línea (es decir, el espacio unidimensional) generado por el vector **w**. Cada punto sobre la esfera unidad se correspondería con una proyección.

3.3.2.2 El algoritmo del gradiente

En la práctica, para maximizar el valor absoluto de la kurtosis, podríamos comenzar tomando un cierto vector **w**, determinando la dirección en la que el valor absoluto de la kurtosis de $y=\mathbf{w}^T\mathbf{z}$ crece de forma más acusada y desplazando **w** en dicha dirección del espacio. Esta es la base del método del gradiente.

A partir de algunas expresiones de interés referentes a la kurtosis [Hyvärinen01] podemos ver que el gradiente del valor absoluto de la kurtosis de $\mathbf{w}^T\mathbf{z}$ se puede determinar como:

$$\frac{\partial\left|kurt(\mathbf{w}^T\mathbf{z})\right|}{\partial w}=4sign(kurt(\mathbf{w}^T\mathbf{z}))\left[E\left\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\right\}-3\mathbf{w}\|\mathbf{w}\|^2\right] \tag{3.23}$$

ya que para los datos blanqueados tenemos que $E\left\{(\mathbf{w}^T\mathbf{z})^2\right\}=\|\mathbf{w}\|^2$. Además como sabemos que estamos optimizando sobre la esfera unidad $\|\mathbf{w}\|^2=1$, el método del gradiente se ha de complementar proyectando **w** sobre dicha esfera en cada paso. Esto se hace tan sólo dividiendo **w** por su norma en cada paso.

Para simplificar más aún dicho algoritmo, podemos ver como el término $3\mathbf{w}\|\mathbf{w}\|^2$ se puede omitir ya que tan sólo podría afectar a la norma de **w** pero no a su dirección. Esto es debido a que realmente sólo interesa la dirección de **w** y cualquier cambio en su norma es poco relevante ya que de todas formas está normalizada a la unidad.

De esta forma obtendremos la siguiente expresión para el algoritmo del gradiente:

$$\begin{gathered}\Delta\mathbf{w}\ \alpha\ sign(kurt(\mathbf{w}^T\mathbf{z}))E\left\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\right\}\\ \mathbf{w}\leftarrow\frac{\mathbf{w}}{\|\mathbf{w}\|}\end{gathered} \tag{3.24}$$

Una versión adaptativa del algoritmo se puede obtener eliminando el operador valor esperado para dar lugar a la siguiente expresión:

$$\Delta \mathbf{w} \propto sign(kurt(\mathbf{w}^T\mathbf{z}))\mathbf{z}(\mathbf{w}^T\mathbf{z})^3$$
$$\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|} \tag{3.25}$$

Por tanto ahora, cada observación **z**(t) se podrá usar en el algoritmo tan sólo una vez. Sin embargo hay que destacar que al evaluar el término $sign(kurt(\mathbf{w}^T\mathbf{z}))$ el valor esperado que aparece en la propia definición de la kurtosis no se puede obviar. Es decir, la kurtosis ha de ser calculada correctamente siguiendo la siguiente regla:

$$\Delta\gamma \propto ((\mathbf{w}^T\mathbf{z})^4 - 3) - \gamma \tag{3.26}$$

siendo γ la estima de la kurtosis.

3.3.2.3 Un algoritmo de punto fijo basado en la kurtosis: FastICA

En esta sección vamos a obtener un algoritmo mucho más eficiente que el descrito en el apartado anterior. En el método del gradiente descrito en previamente presenta el inconveniente fundamental de que su convergencia es muy lenta y presenta una gran dependencia del valor inicial que se elija para comenzar las iteraciones.

Para obtener este nuevo algoritmo hemos de tener en cuenta que en un punto *estable* obtenido a partir del método del gradiente, dicho gradiente ha de estar en la dirección de **w**, es decir, debe ser igual a **w** escalado por una cierta constante. Sólo en este caso, añadiendo el gradiente al vector **w** no se cambiaría su dirección y podríamos alcanzar la convergencia. Este hecho se puede probar recurriendo a la teoría de *Multiplicadores de Lagrange*. De esta forma tendríamos la siguiente ecuación que serviría para **w**:

$$\mathbf{w} \propto \left[E\left\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\right\} - 3\|\mathbf{w}\|^2 \mathbf{w} \right] \tag{3.27}$$

Esta ecuación nos llevará seguidamente a formular un algoritmo de punto fijo que permitiría ir actualizando **w**:

$$\mathbf{w} \leftarrow E\left\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\right\} - 3\mathbf{w} \tag{3.28}$$

Después de cada iteración, **w** se divide por su norma para mantener la restricción referente a la esfera unidad, es decir, $\|\mathbf{w}\| = 1$. Tras obtener el valor definitivo de **w** podremos determinar la componente independiente a estimar con tan sólo efectuar la siguiente operación $\mathbf{w}^T\mathbf{z}$.

Hay que destacar, que la convergencia de este algoritmo significa que el nuevo y el antiguo valor de **w** que se obtenga en cada iteración tendrán la misma dirección, es decir, su producto escalar será igual a uno. No es necesario que el vector converja a un mismo punto ya que tanto **w** como **–w** dan lugar a la misma dirección espacial. Esto

viene motivado de nuevo porque la componente independiente a estimar puede ser determinada tan sólo a falta de una constante multiplicativa.

Todas estas implicaciones provocarán que este algoritmo converja de forma muy *rápida*, por lo que es conocido como algoritmo *FastICA*. Dicho algoritmo presenta un par de propiedades que lo hace más potente que los métodos basados en el gradiente. En primer lugar, se puede demostrar que a convergencia que presenta es *cúbica*, lo cual hace que su convergencia sea así de rápida [Hyvärinen01]. En segundo lugar, al contrario que lo que ocurre en los algoritmos basados en el gradiente, no hay que introducir parámetros que permitan ajustar el algoritmo, lo cual hace que FastICA sea más fácil de manejar.

3.3.2.4 Ejemplo de aplicación del algoritmo FastICA basado en la kurtosis para la estima de una componente independiente

En el primer ejemplo, consideraremos dos distribuciones uniformes entre '-1' y '1' (subgaussianas) que han sido mezcladas con una cierta matriz **A** y blanqueadas a continuación mediante *PCA*. Al aplicar el algoritmo FastICA descrito anteriormente se puede comprobar la rápida convergencia (*5* iteraciones) del vector **w** a su valor definitivo $\mathbf{w} = [0.9859, 0.1675]^T$. Gráficamente lo podemos comprobar en la siguiente figura:

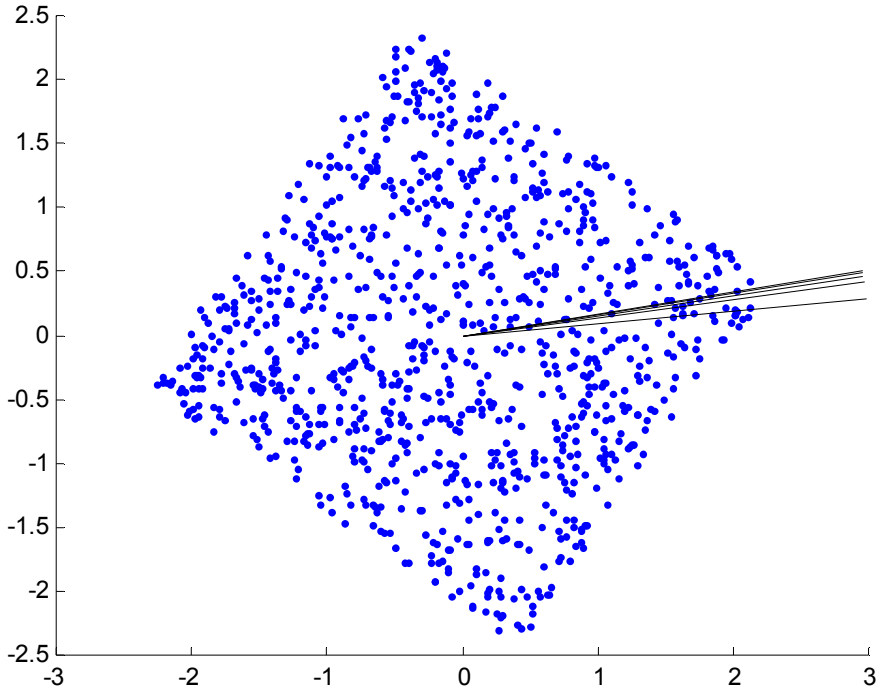

Figura 3.6 *Resultado de aplicar el algoritmo FastICA usando la kurtosis a las distribuciones uniformes originales. Observamos la rápida convergencia del vector* ***w****, representado por las líneas negras.*

Por otro lado, podemos comprobar la evolución del valor de la kurtosis de la proyección $\mathbf{w}^T\mathbf{z}$ durante las cinco iteraciones que realizó el algoritmo. Como era de esperar, por ser una distribución subgaussiana, la kurtosis tiende hacia un valor negativo y mayor que '-2':

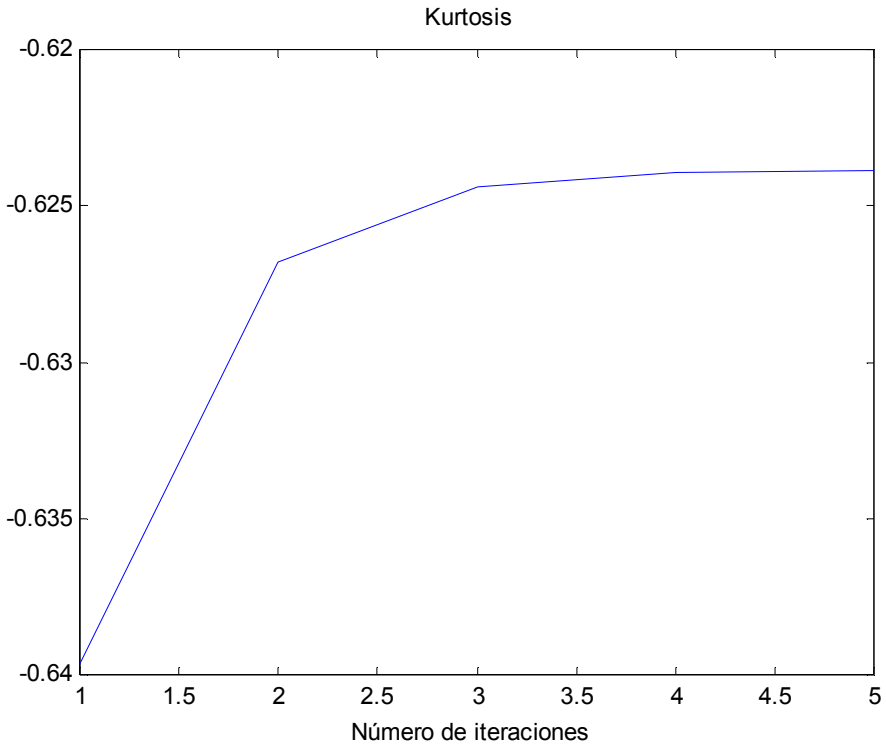

Figura 3.7 *Evolución de la kurtosis de* $\mathbf{w}^T\mathbf{z}$ *durante las cinco iteraciones de las que constó la simulación. Por tratarse de variables subgaussianas, dicho valor tiende a una cantidad negativa y superior a '-2'.*

En este ejemplo, hemos estimado tan sólo una componente independiente, pero a veces es necesario estimar más de una. Para verlo de forma gráfica tan sólo tendremos que fijarnos en la dirección que es ortogonal al vector **w**, que nos permitirá obtener la segunda y última componente independiente a estimar en este ejemplo. En la siguiente gráfica se muestra la dirección que tendría el vector **w** que daría lugar a la segunda componente independiente al hacer $\mathbf{w}^T\mathbf{z}$:

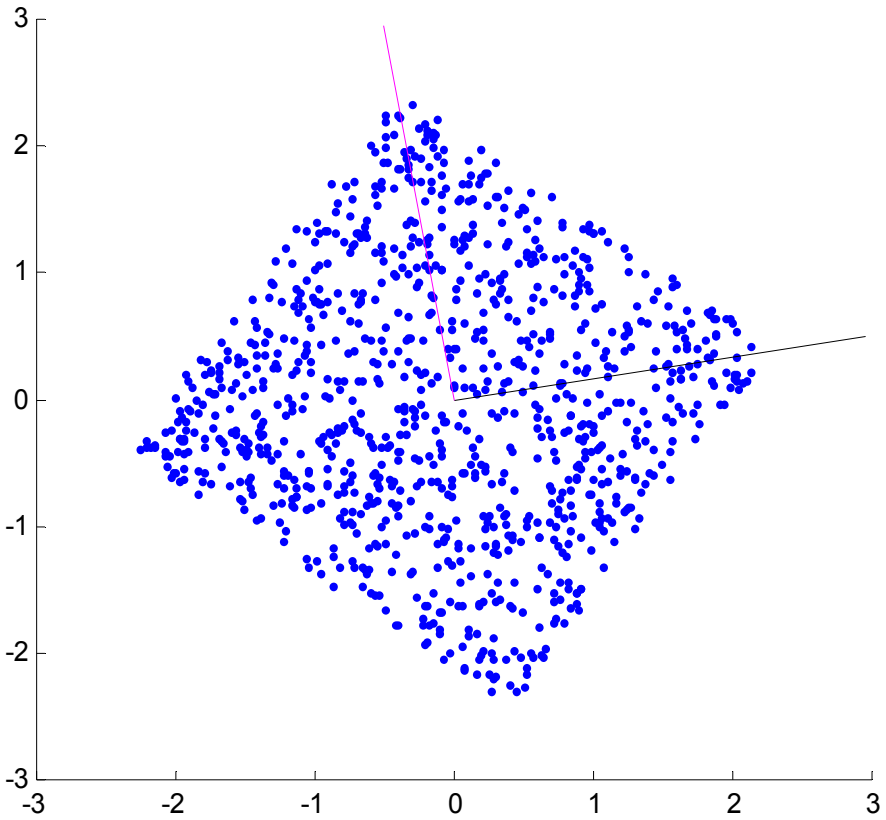

Figura 3.8 *Obtención del segundo vector* ***w*** *(en magenta) que daría lugar a la segunda componente independiente a estimar. Como se observar en la figura, dicho vector es ortogonal al obtenido previamente (en negro).*

El estudio del algoritmo FastICA para la estima de más de dos componentes será objeto de estudio en apartados posteriores.

3.3.3 El método del gradiente y la entropía negativa

En el apartado anterior vimos que mediante la kurtosis podíamos disponer de un método sencillo para estima el modelo *ICA*. Sin embargo la kurtosis presenta algunos problemas en la práctica a la hora de estimar correctamente la no gaussianidad, como puede ser el caso de su extrema sensibilidad. Por el contrario, otras medidas de la no gaussianidad funcionan mejor en ciertas situaciones, como puede ser el caso de la *entropía negativa*, que desarrollaremos en este apartado. Sus propiedades son contrarias a las de la kurtosis, es decir, presenta mayor robustez pero su cálculo implica una mayor carga computacional, aunque veremos también algunas aproximaciones que harán que su determinación se simplifique considerablemente [Hyvärinen01].

3.3.3.1 Aproximando la entropía negativa

La entropía de una variable aleatoria está relacionada con la información que dicha variable aporta, de forma que cuanto mayor sea su entropía, mayor será su incertidumbre (mayor información nos dará). Como ya vimos en apartados anteriores, la entropía diferencial *H* de un vector aleatorio **y** con densidad $p_y(\xi)$ se define como:

$$H(\mathbf{y}) = -\int p_y(\xi)\log p_y(\xi)d\xi \tag{3.29}$$

Para extraer una medida de la no gaussianidad se usa la *entropía negativa J*, que no es más que una versión normalizada de la entropía diferencial *H*:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - \mathrm{H}(\mathbf{y}) \tag{3.30}$$

Una vez establecidas las definiciones básicas, vamos a estudiar una forma de aproximar escalarmente la entropía negativa mediante estadísticos de orden superior y usando una expansión polinomial basada en la de *Gram-Charlier* [Hyvärinen01]:

$$J(y) \approx \frac{1}{12}E\left\{y^3\right\}^2 + \frac{1}{48}kurt(y)^2 \tag{3.31}$$

donde *y* se supone que es de media cero y de varianza unidad. Una aproximación de gran utilidad consiste en generalizar la aproximación basada en los estadísticos de orden superior de forma que se usen valores esperados de funciones no cuadráticas. En general, podremos intercambiar las funciones y^3 e y^4 por otras funciones G^i (donde *i* es un índice, no una potencia), posiblemente más de dos. Este método proporcionará por tanto una manera muy simple de aproximar la entropía negativa basándonos en los valores esperados de las $G^i(y)$. En el caso que nos ocupa, referido a la entropía negativa, veremos como es posible tomar dos funciones no cuadráticas, que denotaremos por G^1 y G^2 de forma que G^1 es impar y es G^2 par, de forma que podremos llegar a la siguiente aproximación:

$$J(y) \approx k_1(E\left\{G^1(y)\right\})^2 + k_2(E\left\{G^2(y)\right\} - E\left\{G^2(\upsilon)\right\})^2 \tag{3.32}$$

donde k_1 y k_2 son dos constantes positivas y υ es una variable gaussiana de media cero y varianza unidad. Hay que destacar que incluso en los casos en los que la aproximación no es muy buena la ecuación anterior se puede emplear como una medida de la no gaussianidad que sea consistente en el sentido de que siempre es no negativa e igual a cero si la variable *y* tiene una distribución gaussiana. Esta será pues una generalización de una aproximación basada en los momentos que se ha obtenido tomando $G^1(y)=y^3$ y $G^2(y)=y^4$.

En el caso de que sólo usemos una función no cuadrática *G* la aproximación quedaría como:

$$J(y)\,\alpha\left[E\left\{G(y)\right\} - E\left\{G(\upsilon)\right\}\right]^2 \tag{3.33}$$

que será válida para prácticamente cualquier función no cuadrática *G*.

Una vez realizada la aproximación, el nuevo problema que se plantea es el de obtener *G* de forma adecuada para que la estimación sea lo mejor posible. Se puede demostrar que para las siguientes funciones *G* los resultados obtenidos fueron considerablemente aceptables:

$$G_1(y) = \frac{1}{a_1}\log\cosh a_1 y$$
$$G_2(y) = -\exp(-\frac{y^2}{2}) \qquad (3.34)$$

donde $1 \le a_1 \le 2$.

Dichas funciones aparecen representadas en la siguiente figura:

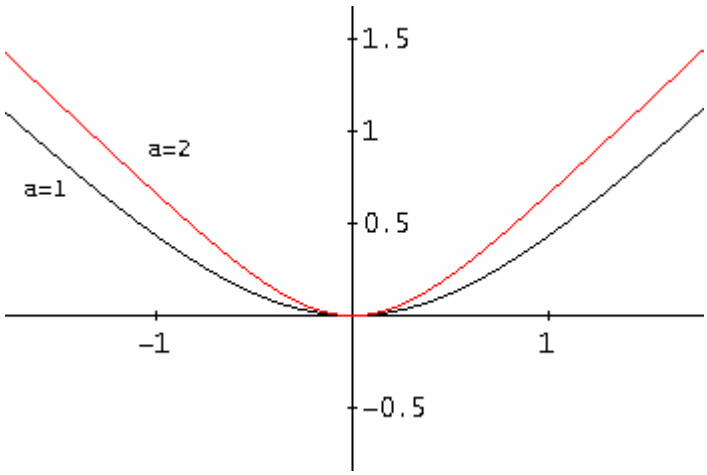

Figura 3.9 *Representación de $G_1(y)$ para los valores dos valores de la constante arbitraria 'a'.*

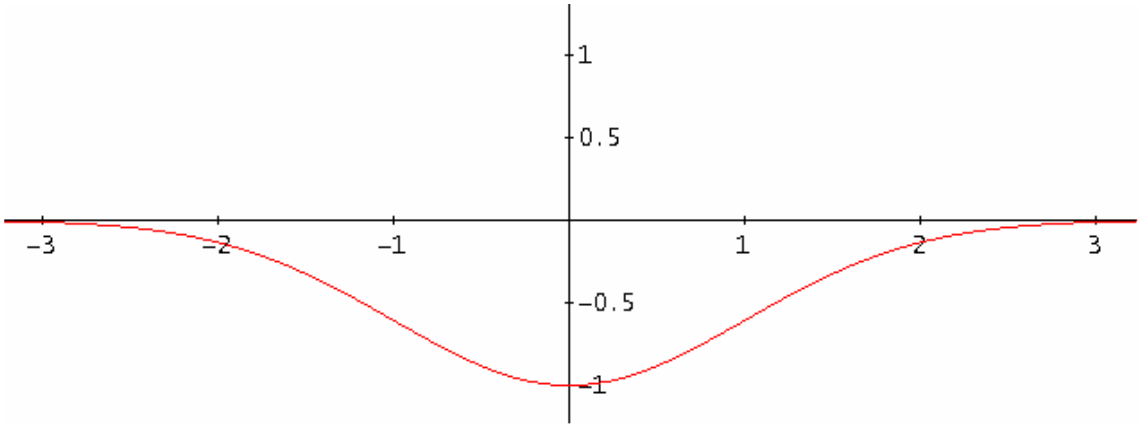

Figura 3.10 *Representación de $G_2(y)$.*

De esta forma, obtendremos aproximaciones de la entropía negativa que ofrecen un buen compromiso entre facilidad de cálculo y sensibilidad.

3.3.3.2 El algoritmo del gradiente basado en la entropía negativa

Al igual que ocurre en el caso de la kurtosis, es posible obtener una versión del algoritmo del gradiente que permita maximizar la entropía negativa. Tomando el gradiente de la ecuación (3.33) respecto a **w** teniendo en cuenta la normalización $E\left\{(\mathbf{w}^T\mathbf{z})^2\right\} = \|\mathbf{w}\|^2 = 1$ obtendremos las siguientes expresiones para el algoritmo:

$$\begin{aligned} \Delta\mathbf{w} &\propto \gamma E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} \\ \mathbf{w} &\leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|} \end{aligned} \tag{3.35}$$

donde se tiene que $\gamma = E\left\{G(\mathbf{w}^T\mathbf{z})\right\} - E\left\{G(\upsilon)\right\}$, siendo υ una variable gaussiana de media cero y varianza unidad. La normalización es necesaria para proyectar **w** en la esfera unidad para así mantener la varianza de $\mathbf{w}^T\mathbf{z}$ constante. La función *g* es la derivada de la función *G* usada en la aproximación de la entropía negativa.

La constante γ que permite aportar al algoritmo ciertas propiedades de 'auto-adaptación' se puede estimar de la siguiente manera:

$$\Delta\gamma \propto (G(w^T z)) - E\left\{G(\upsilon)\right\} - \gamma \tag{3.36}$$

Por otro lado veremos que para obtener las funciones *g* apropiadas para este algoritmo, tendremos que recurrir a las derivadas de las funciones *G* que aparecen en la ecuación (3.34). De forma alternativa, podríamos emplear las derivadas correspondientes a la cuarta potencia, tal y como se hace en el método de la kurtosis, de forma que las funciones *g* quedarían de la siguiente manera:

$$\begin{aligned} g_1(y) &= \tanh(a_1 y) \\ g_2(y) &= y\exp(-\frac{y^2}{2}) \\ g_3(y) &= y^3 \end{aligned} \tag{3.37}$$

siendo $1 \le a_1 \le 2$ una constante, que generalmente se suele tomar como la unidad.

En las siguientes figuras podemos observar la forma que toman las funciones *g* expuestas en la ecuación (3.37):

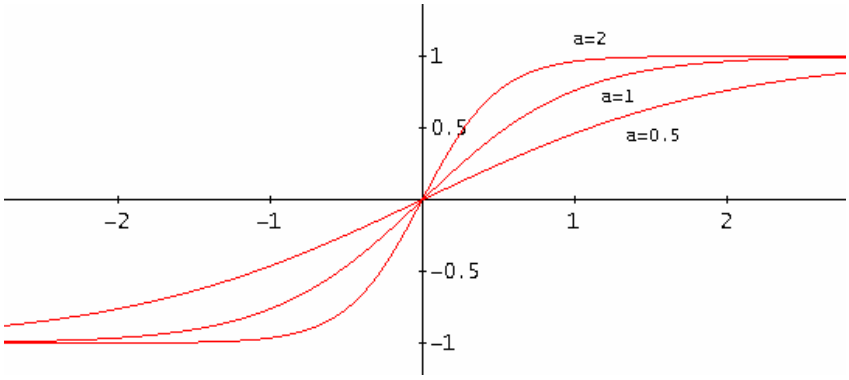

Figura 3.11 *Representación de $g_1(y)$ para diferentes valores de 'a'*

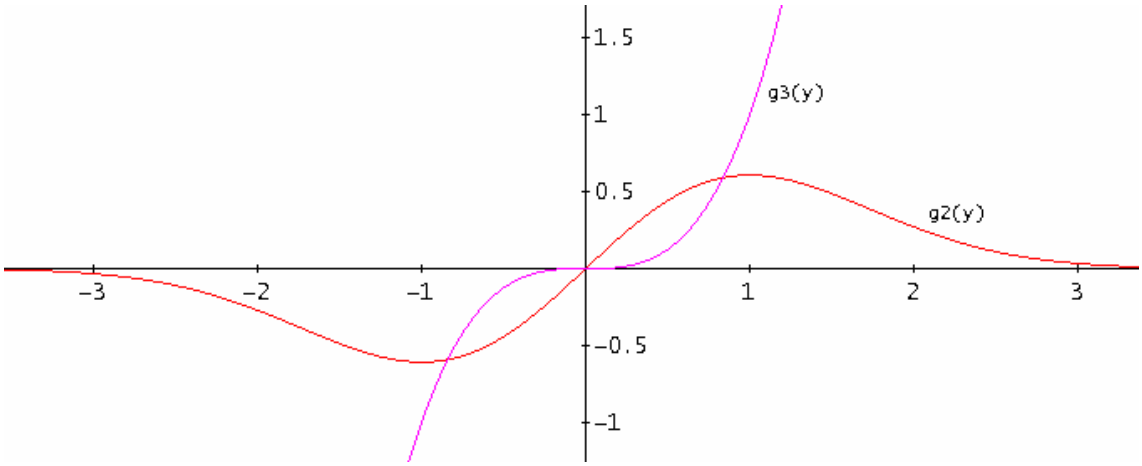

Figura 3.12 *Representaciones de $g_2(y)$ (en rojo) y $g_3(y)$ (en magenta)*

Con todo, podemos resumir los pasos que hemos dado en el algoritmo en el siguiente esquema:

1. Centrar los datos para hacer que tengan media cero.
2. Blanquear los datos para obtener **z**.
3. Tomar un vector **w** inicial de norma unidad y un valor inicial para γ.
4. Actualizar $\Delta \mathbf{w} \propto \gamma \mathbf{z} g(\mathbf{w}^T \mathbf{z})$ eligiendo g según la ecuación (3.37).
5. Normalizar **w**, es decir, hacer $\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|}$.
6. Si el signo de γ no es conocido a priori actualizar $\Delta\gamma \propto (G(w^T z)) - E\{G(\upsilon)\} - \gamma$.
7. Si no converge, retomar el paso 4.

3.3.3.3 Un algoritmo de punto fijo basado en la entropía negativa: FastICA

Como ocurre en el caso de la kurtosis, es posible definir un método mucho más rápido para maximizar la entropía negativa que el que proporciona el método del gradiente, se trata del algoritmo *FastICA*. Este algoritmo trata de encontrar un vector **w** tal que la proyección $\mathbf{w}^T\mathbf{z}$ maximice la no gaussianidad. La no gaussianidad se

determina en este caso por la aproximación de la entropía no negativa dada en la ecuación (3.33). Hay que tener en cuenta además que la varianza de $\mathbf{w}^T\mathbf{z}$ ha de ser la unidad, es decir, para los datos blanqueados es equivalente a afirmar que la norma de **w** es uno [Hyvärinen01].

La justificación de la expresión que toma el algoritmo FastICA en este caso, se basará fundamentalmente en el *método iterativo de Newton*. A partir de la ecuación (3.35) relativa al método del gradiente basado en la entropía no negativa, podemos deducir la siguiente iteración de punto fijo:

$$\begin{aligned} &\mathbf{w} \leftarrow E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\}. \\ &\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|} \end{aligned} \tag{3.38}$$

La expresión anterior no presenta las buenas prestaciones en cuanto a convergencia que tiene FastICA basado en la kurtosis, ya que los momentos no polinomiales no poseen las mismas propiedades algebraicas óptimas de la kurtosis. Por tanto la iteración de la ecuación anterior tiene que ser modificada. Esto es posible ya que podemos añadir el vector **w** escalado por una cierta constante α en ambos miembros de la ecuación (3.38):

$$\begin{aligned} \mathbf{w} = E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} \Leftrightarrow (1+\alpha)\mathbf{w} = E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} + \alpha\mathbf{w} \\ \mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|} \end{aligned} \tag{3.39}$$

La elección de α es crítica, ya que si se toma de forma óptima podremos hacer que el algoritmo FastICA basado en la entropía negativa, converja tan rápidamente como lo hace el basado en la kurtosis.

El método de Newton aplicado al gradiente proporciona un método de optimización que converge en pocas iteraciones. Sin embargo, el problema de este método radica en que requiere una inversión de matriz en cada paso por lo que la carga computacional final no difiere en exceso de la que necesitan los métodos basados en el gradiente. Sin embargo es posible hacer uso de las propiedades que presenta el problema *ICA* para encontrar una aproximación del método de Newton que no necesita de esa inversión de la matriz pero que converge en el mismo número de iteraciones que el método original. Este método dará un algoritmo como el de la ecuación (3.39).

Para justificar el método de Newton aproximado, debemos basarnos en la teoría de *multiplicadores de Lagrange* aplicada al método de Newton. Tras desarrollar una serie de operaciones intermedias descritas en [Hyvärinen01] llegaremos a una expresión cerrada que representará el algoritmo FastICA deseado:

$$\mathbf{w} \leftarrow E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z}) - E\left\{g'(\mathbf{w}^T\mathbf{z})\right\}\right\} \tag{3.40}$$

siendo *g'* la derivada de la función *g*.

Con todo, podemos resumir los pasos que hemos dado en el algoritmo en el siguiente esquema:

1. Centrar los datos para hacer que tengan media cero.
2. Blanquear los datos para obtener **z**.
3. Tomar un vector **w** inicial de norma unidad.
4. Actualizar $\mathbf{w} \leftarrow E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} - E\left\{g'(\mathbf{w}^T\mathbf{z})\right\}\mathbf{w}$ con g según la ecuación (3.37).
5. Normalizar **w**, es decir, hacer $\mathbf{w} \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|}$.
6. Si no hay convergencia, retornaremos al paso 4.

Llegados a este punto, nos planteamos la elección de la función *g'*. Tomando como base el conjunto de funciones definidas en la ecuación (3.37), llegaremos a que:

$$\begin{aligned} g_1'(y) &= a_1(1-\tanh^2(a_1 y)) \\ g_2'(y) &= (1-y^2)\exp(-\frac{y^2}{2}) \\ g_3' &= 3y^2 \end{aligned} \tag{3.41}$$

3.3.4 Estima de más de una componente independiente

En los apartados anteriores hemos descrito técnicas que permiten tan sólo estimar una sola componente independiente, por lo que los algoritmos anteriores son conocidos como algoritmos *"one-unit"*. En principio, sería posible estimar un número mayor de componentes independientes tan sólo ejecutando el algoritmo un número mayor de veces y haciendo uso de diferentes puntos iniciales. De todas formas esta forma de proceder no sería un método muy apropiado cuando el número de componentes independientes a estimar es muy elevado [Hyvärinen01].

La idea en la que se fundamenta la extensión del método basado en la maximización de la no gaussianidad se fundamenta en la propiedad de que los vectores $\boldsymbol{w_i}$ correspondientes a diferentes componentes independientes son *ortogonales en el espacio "blanqueado"*. Es decir, la independencia de las componentes independientes requiere que estén incorreladas, además en el espacio resultante tras el blanqueo tendremos que $E\left\{\left(\mathbf{w}_i^T\mathbf{z}\right)\left(\mathbf{w}_j^T\mathbf{z}\right)\right\} = \mathbf{w}_i^T\mathbf{w}_j$ por lo que la incorrelación es equivalente a la ortogonalidad.

Esta propiedad es consecuencia directa del hecho de que después del blanqueo la matriz de mezcla se puede considerar que es ortogonal. Los $\boldsymbol{w_i}$ por definición son las filas de la inversa de la matriz de mezcla y éstas son iguales a las columnas de la matriz de mezcla debido a que por la propiedad de ortogonalidad, $\mathbf{A}^{-1} = \mathbf{A}^T$.

De esta forma, para estimar diferentes componentes independientes necesitamos ejecutar varias veces el algoritmo para estimar una sola componente con vectores

$\mathbf{w}_1, ..., \mathbf{w}_n$ y para evitar que dos o más vectores diferentes converjan al mismo máximo, hemos de ortogonalizar los $\mathbf{w}_1, ..., \mathbf{w}_n$ después de cada iteración del algoritmo. En los siguientes apartados se desarrollarán diferentes métodos que permitan alcanzar la incorrelación que requiere el método.

3.3.4.1 Ortogonalización deflacionaria

Esta forma de ortogonalización se basa en el *método de Gram-Schmidt*, y se fundamenta en ir estimando las componentes independientes una a una. Cuando hayamos estimado *p* componentes independientes o *p* vectores $\mathbf{w}_1, ..., \mathbf{w}_p$, se ejecuta el algoritmo para la estima de una sola componente para estimar $\mathbf{w}_{p+1}$ y después de cada iteración, se eliminan de $\mathbf{w}_{p+1}$ las proyecciones $\left(\mathbf{w}_{p+1}^T \mathbf{w}_j\right)\mathbf{w}_j, j = 1, ..., p$ de los *p* vectores previamente estimados. Posteriormente habrá que normalizar $\mathbf{w}_{p+1}$.

El método de ortogonalización se puede resumir en los siguientes pasos:

1. Elegimos el número de componentes independientes a estimar, *m*. Inicializamos *p* a uno.
2. Inicializamos $\mathbf{w}_p$.
3. Ejecutamos el algoritmo para la estima de una componente independiente tomando $\mathbf{w}_p$.
4. Se realiza la siguiente ortogonalización:

$$w_p \leftarrow w_p - \sum_{j=1}^{p-1} \left(\mathbf{w}_p^T \mathbf{w}_j\right)\mathbf{w}_j \tag{3.42}$$

5. Normalizamos $\mathbf{w}_p$.
6. Si $\mathbf{w}_p$ no converge, volvemos al paso 3.
7. Aumentar *p* en una unidad y si no es mayor que el número de componentes independientes a estimar, volvemos al paso 2.

3.3.4.2 Ortogonalización Simétrica

En ciertas aplicaciones, sería deseable hacer uso de una decorrelación simétrica donde no haya ningún vector que tenga más importancia que el resto. Esto significa que los vectores $\mathbf{w}_i$ no se estiman una a una sino que se hace de forma paralela para todos ellos.

Un motivo para elegir este método es que el método anterior tiene el inconveniente es que el error de la estimación en los primeros vectores se acumula para los siguientes debido a la ortogonalización. Otro es que la ortogonalización simétrica permite el cálculo paralelo de las componentes independientes.

La ortogonalización simétrica se puede resumir en los siguientes pasos:

1. Elegimos el número de componentes independientes a estimar, *m*.

2. Inicializamos $\mathbf{w}_i, i = 1, ..., m$.
3. Ejecutamos el algoritmo para la estima de una componente independiente para los $\mathbf{w}_i$ en paralelo.
4. Realizar una ortogonalización simétrica de la matriz $\mathbf{W} = (\mathbf{w_1}, ..., \mathbf{w_m})^T$.
5. Si no hay convergencia, retornaremos al paso 3.

Por otro lado podemos ver como es posible definir una forma alternativa de formular el algoritmo anterior [Hyvärinen01]. Su formulación será de la siguiente manera:

1. Normalizando $\mathbf{W} \leftarrow \frac{\mathbf{W}}{\|\mathbf{W}\|}$.
2. Actualizamos **W** según la siguiente regla:

$$\mathbf{W} \leftarrow \frac{3}{2}\mathbf{W} - \frac{1}{2}\mathbf{W}\mathbf{W}^T\mathbf{W} \tag{3.43}$$

3. Si $\mathbf{W}\mathbf{W}^T$ no está lo suficientemente próxima a la matriz identidad, volvemos al paso 2.

La norma calculada en el *paso 1* puede tomarse como cualquier norma típica en el cálculo matricial (pero no la norma de Frobenius).

3.3.4.3 Ejemplo de estimación de tres componentes independientes mediante el algoritmo FastICA con ortogonalización deflacionaria

En el siguiente ejemplo vamos a ver como funciona el algoritmo FastICA cuando queremos estimar tres componentes independientes. Para comprobarlo, supondremos que las tres componentes independientes a estimar se corresponden con señales aleatorias bipolares, es decir, que sólo pueden tomar valores '-1' o '1'.

Estas señales se muestran en la siguiente figura:

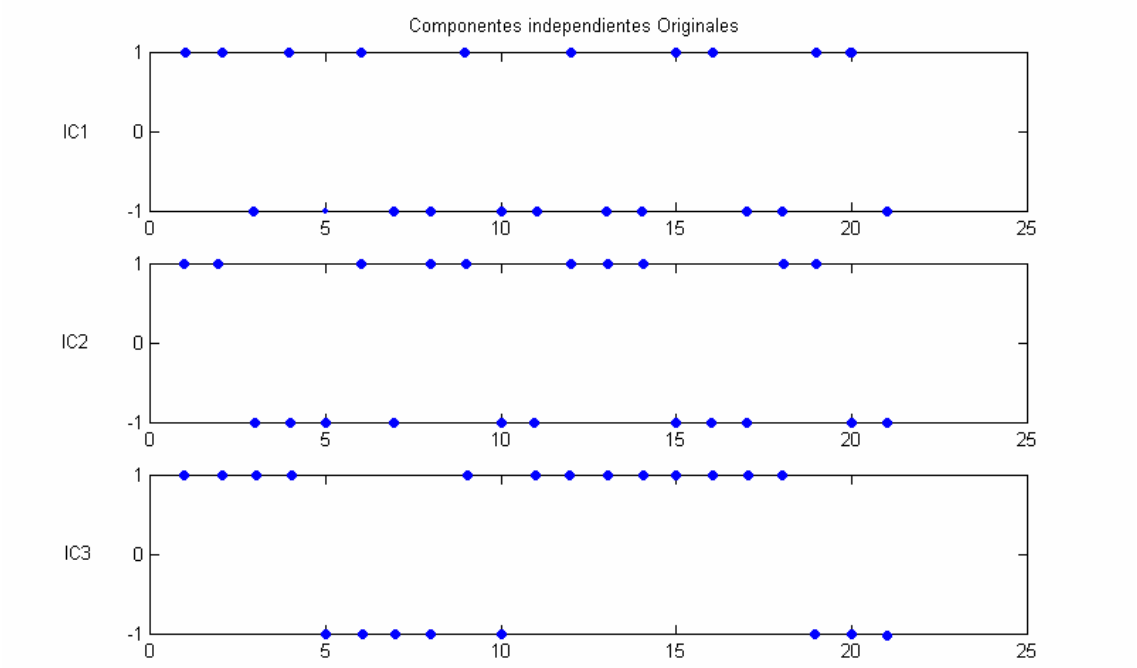

Figura 3.13 Las *componentes independientes originales a estimar se corresponden con tres señales bipolares.*

Al aplicar el algoritmo FastICA para las tres componentes independientes se obtuvieron los siguientes resultados, que sin duda se corresponden con las señales originales, salvo una constante multiplicativa (como ya estudiamos previamente), que en este caso será '-1'.

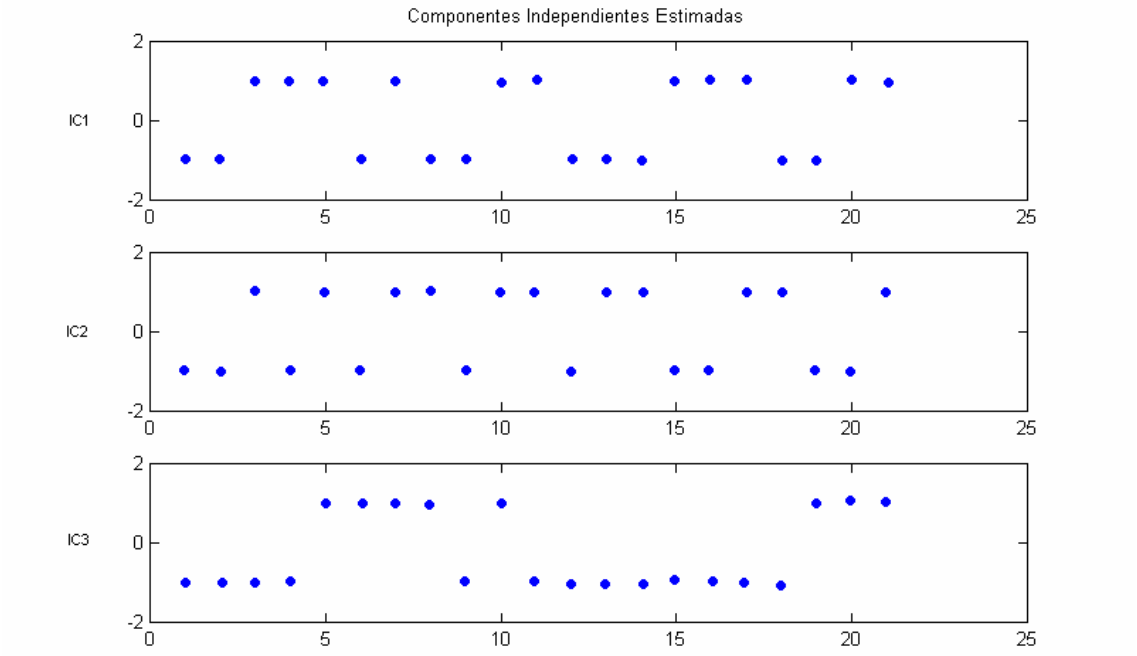

Figura 3.14 *Componentes independientes estimadas. Cabe destacar que como se ha venido estudiando a lo largo del capítulo, las señales estimadas pueden variar de las originales en una constante multiplicativa (en este caso '-1').*

3.4 Estimación del modelo *ICA* a partir de la matriz de covarianzas

Concluimos este capítulo dedicado al desarrollo de diferentes métodos de estimación de la transformación *ICA* estudiando un método basado en la estructura temporal de las señales. En este apartado, consideraremos que el modelo a estimar, las componentes independientes son señales en el tiempo, es decir, $s_i(t), t = 1, ..., T$, siendo t el índice temporal.

Por tanto, el modelo quedará resumido en la siguiente expresión:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{3.44}$$

donde **A** se considera una matriz cuadrada y las componentes serán independientes. Sin embargo, la principal diferencia radica en que a diferencia de lo que se ha estudiado hasta ahora, las componentes independientes no tienen que ser necesariamente gaussianas.

A continuación estableceremos algunas hipótesis sobre la estructura temporal de las componentes independientes que permitirán obtener el modelo *ICA*. Estás hipótesis serían en cierto modo equivalentes a las referentes a la no gaussianidad que se han venido enunciando hasta ahora.

En primer lugar, se podría considerar que las componentes independientes tienen diferentes autocovarianzas (y que en general serán distintas de cero). Y en segundo lugar, sería posible afirmar que las varianzas de las componentes independientes son no estacionarias. En este apartado, estudiaremos sólo la referente a las autocovarianzas.

3.4.1 Estimación basada en la covarianza

La forma más simple de definir una estructura temporal es a partir de las autocovarianzas, es decir, las covarianzas entre los valores de una misma señal para diferentes instantes de tiempo:

$$\mathrm{cov}(x_i(t)x_i(t-\tau)) \tag{3.45}$$

siendo τ una constante que significa un *retraso* $\tau = 1, 2, 3, \ldots$.

Además de las autocovarianzas de una señal, necesitaremos las covarianzas entre dos señales diferentes, es decir:

$$\mathrm{cov}(x_i(t)x_j(t-\tau)) \text{ con i} \neq \text{ j} \tag{3.46}$$

Ambas expresiones se pueden agrupar en una misma matriz de covarianzas que denotaremos como $\mathbf{C}_\tau^x$ y que tendrá la siguiente expresión:

$$\mathbf{C}_{\tau}^{x} = E\left\{\mathbf{x}(t)\mathbf{x}(t-\tau)^{T}\right\} \tag{3.47}$$

La clave que permite resolver el problema está en que la información contenida en la matriz de covarianzas $\mathbf{C}_{\tau}^{x}$ se puede usar en vez de los estadísticos de orden superior. Una forma de verlo es considerar que las covarianzas entre señales diferentes sean cero, lo cual va a llevar a que las componentes sean *independientes*:

$$E\left\{y_i(t)y_j(t-\tau)\right\} = 0 \text{ para todo i} \neq \text{j, j, } \tau \tag{3.48}$$

3.4.1.1 Estimación de la transformación usando un solo retraso

Es el caso más simple y el que será objeto de nuestro estudio. Consideraremos la existencia de un solo retraso τ que se suele tomar como la unidad. A partir de aquí, podremos formular un algoritmo muy simple que permitirá resolver nuestro problema.

Llamando $\mathbf{z}$ a los datos procedentes del proceso de blanqueo, las ecuaciones del modelo resultan de la siguiente forma:

$$\begin{aligned}\mathbf{W}\mathbf{z}(t) &= \mathbf{s}(t)\\ \mathbf{W}\mathbf{z}(t-\tau) &= \mathbf{s}(t-\tau)\end{aligned} \tag{3.49}$$

Consideremos ahora una ligera variación de la matriz de covarianzas descrita en la ecuación (3.47):

$$\overline{\mathbf{C}}_{\tau}^{z} = \frac{1}{2}\left[\mathbf{C}_{\tau}^{z} + \left(\mathbf{C}_{\tau}^{z}\right)^{T}\right] \tag{3.50}$$

de forma que teniendo en cuenta las propiedades de linealidad y ortogonalidad:

$$\overline{\mathbf{C}}_{\tau}^{z} = \frac{1}{2}\mathbf{W}^{T}\left[E\left\{\mathbf{s}(t)\mathbf{s}(t-\tau)^{\mathrm{T}}\right\} + E\left\{\mathbf{s}(t-\tau)\mathbf{s}(t)^{\mathrm{T}}\right\}\right]\mathbf{W} = \mathbf{W}^{T}\overline{\mathbf{C}}_{\tau}^{s}\mathbf{W} \tag{3.51}$$

Debido a la independencia de las $s_i(t)$, la matriz $\mathbf{C}_{\tau}^{s} = E\left\{\mathbf{s}(t)\mathbf{s}(t-\tau)\right\}$ es una matriz diagonal, que de ahora en adelante llamaremos **D**. Por su estructura, vemos que $\overline{\mathbf{C}}_{\tau}^{s}$ es igual a $\mathbf{C}_{\tau}^{s}$. De esta forma tendremos que:

$$\overline{\mathbf{C}}_{\tau}^{z} = \mathbf{W}^{T}\mathbf{D}\mathbf{W} \tag{3.52}$$

La ecuación anterior trata de mostrar que la matriz **W** forma parte de la descomposición en autovalores de $\overline{\mathbf{C}}_{\tau}^{z}$.

3.4.1.2 El algoritmo AMUSE

Todo el desarrollo expuesto anteriormente, se puede resumir en un algoritmo matemático, llamado *AMUSE* que permite obtener la matriz **W**. Dicho método se puede resumir en los siguientes pasos:

1. Blanqueo de los datos **x** para obtener *z(t)*.
2. Realizar la descomposición en autovalores y autovectores de la matriz $\bar{\mathbf{C}}_{\tau}^{z} = \frac{1}{2}\left[\mathbf{C}_{\tau}^{z} + \left(\mathbf{C}_{\tau}^{z}\right)^{T}\right]$.
3. Las filas de la matriz de separación **W** vienen dadas por los autovectores de $\bar{\mathbf{C}}_{\tau}^{z}$.

Este algoritmo en muy rápido y simple de ejecutar. Sin embargo presenta como principal problema que su funcionamiento no es el adecuado cuando existen autovalores que son iguales entre si. En ese caso, las componentes independientes no pueden ser estimadas. Una forma de solucionar este problema podría consistir en encontrar un valor de τ que haga que todos los autovalores sean distintos, si bien esto no es siempre factible. Si las señales $s_i(t)$ tienen idénticas autocovarianzas, no es posible encontrar ningún valor de τ que permita estimar las componentes independientes.

3.5 Conclusiones

En este segundo capítulo dedicado al Análisis de Componentes Independientes se han presentado diferentes técnicas que permiten separar las señales a estimar. En primer lugar, destacamos algunos aspectos referentes al preprocesado y blanqueo de los datos antes de aplicar la transformación ICA.

A continuación consideramos algunos métodos de estimación de las componentes independientes. En primer lugar, se realizó un estudio basado en la maximización de la no gaussianidad como objetivo para llegar a las condiciones óptimas de separación. Esta técnica se puede dividir a su vez en dos formas de proceder según empleemos como herramienta matemática para implementarla la *kurtosis* o bien principios basados en la Teoría de la Información, como es el caso de la *entropía negativa*. En ambos casos se desarrolló un algoritmo de punto fijo de gran utilidad de cara a su realización en lenguajes de simulación matemática, que es el conocido como FastICA. Por último vimos como todo el estudio que se había realizado hasta el momento, referente a la estima de componentes independientes, se puede hacer extensible a un número mayor con tan sólo aplicar alguno de los dos métodos de ortogonalización (deflacionaria o simétrica) descritos.

Por último, se expuso el algoritmo AMUSE, un método de separación basado en las covarianzas de las señales implicadas en el proceso de separación. Este método destaca sobre todo por su enorme facilidad de implementación y por ofrecer unos resultados muy aceptables, si bien su principal inconveniente radica en que deja de funcionar correctamente cuando se da el caso de igualdad entre algunos de los autovalores de la matriz de covarianzas.