

# ESTIMACIÓN DE FUNCIONES NO LINEALES EN MEZCLAS POST-NO LINEALES

Jordi Solé-Casals<sup>(1)</sup>, Christian Jutten<sup>(2)</sup>

Jordi.Sole@uvic.cat, Christian.Jutten@inpg.fr

<sup>(1)</sup> Grup de Processament de Senyal, Universitat de Vic, c/ de la Laura, 13, 08500 Vic

<sup>(2)</sup> GIPSA Lab., INPG, 46 Av.Félix Viallet, 38031 Grenoble (France)

**Abstract-** This paper proposes a new method for blindly inverting a nonlinear mapping which transforms a sum of random variables. This is the case of post-nonlinear (PNL) source separation mixtures. The importance of the method is based on the fact that it permits to decouple the estimation of the nonlinear part from the estimation of the linear one. Only the nonlinear part is inverted, without considering on the linear part. Hence the initial problem is transformed into a linear one that can then be solved with any convenient linear algorithm. The method is compared with other existing algorithms for blindly approximating nonlinear mappings. Experiments show that the proposed algorithm outperforms the results obtained with other algorithms and give a reasonably good linearized data

## I. INTRODUCCIÓN

La separación ciega de modelos post-no lineales (PNL) (ver Fig. 1) requiere la estimación de la inversa de la función no lineal y de la parte lineal (matriz de mezcla). Esto se puede realizar minimizando la información mutua de la salida de la estructura de inversión. Sin embargo, nos conduce a algoritmos lentos, puesto que las dos partes (no lineal y lineal) están en cascada y se optimizan con el mismo criterio.

Un método simple y muy rápido para estimar de forma aproximada la inversa de la función no lineal se presenta en [1]. El método proporciona un buen punto de la inicialización que se puede utilizar como punto de partida de cualquier algoritmo para invertir la mezcla PNL. Esto no modifica el funcionamiento asintótico del algoritmo utilizado y aumenta su velocidad de convergencia, con un coste computacional adicional muy pequeño.

En este artículo, proponemos una generalización de esta idea, explotada ya en [1][4], para estimar la inversa de la función no lineal. Esta estimación permite transformar el caso PNL a un problema lineal, que puede ser invertido con cualquier algoritmo lineal.

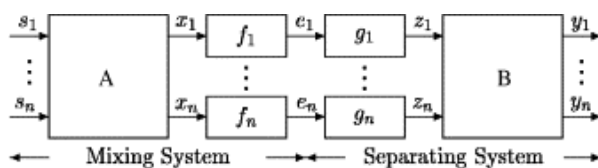


Fig. 1: El sistema de mezcla-separación para el caso PNL

Generalizando resultados anteriores, proponemos un nuevo algoritmo para la estimación de la inversa de la función no lineal  $f_i$ .

El algoritmo consiste en el ajuste de una función no lineal  $g_i$  de modo que la entropía de Shannon de  $Z_i=g_i(E_i)$  sea máxima bajo la condición de variancia unitaria. Aunque parece que estamos haciendo lo mismo que [1], aquí la variable aleatoria recuperada  $z$  (después de que la inversión no lineal) no tiene forzosamente una distribución Gaussiana pero sí una entropía máxima. La sección 2 explica los principios del algoritmo. La sección 3 muestra experimentalmente la robustez y el funcionamiento del método, y finalmente se presentan las conclusiones en la sección 4.

## II. ALGORITMO

Considerar la entropía de Shannon de la variable aleatoria continua con variancia unidad  $Z = g(E)$ :

$$H(Z) = \int -\log p_z(u) p_z(u) du \quad (1)$$

Donde  $p_z(u)$  denota la función de densidad de probabilidad. Dado que la variancia es unitaria, la entropía de Shannon  $H(Z)$  es máxima si  $Z$  es Gaussiana [2], con lo cual  $g$  se puede estimar de manera que  $H(Z)=H(g(E))$  sea máxima (bajo la condición de variancia unitaria).

En el caso discreto, las ecuaciones anteriores quedan como sigue:

$$H(Z) = -\sum_{k=1}^N \log p(z_k) p(z_k) \quad (2)$$

Donde  $p(z_k)$  son las probabilidades de un conjunto de mensajes  $\{z_1, \dots, z_n\}$  ocurriendo con probabilidades  $p_1, \dots, p_n$  [2].

Dado que para variables continuas la entropía máxima para una variancia determinada corresponde a una distribución Gaussiana y que para el caso discreto corresponde a una distribución uniforme, el nuevo método que proponemos debe obtener buenos resultados en los dos casos, ya que no forzamos directamente la obtención de un tipo u otro de función de distribución para  $Z$  sino que maximizamos la entropía  $H(Z)$ . Los resultados experimentales confirman este punto cuando comparamos

los tres algoritmos (Gaussianización, uniformización y maximización de entropía).

La parametrización más simple consiste en utilizar un polinomio. Esto nos dará un algoritmo sencillo y con pocos parámetros.

Consideremos un polinomio de grado  $N$  para  $g(\cdot)$ :

$$g(t) = \sum_{n=1}^{N+1} a_n t^{n-1} \quad (3)$$

De acuerdo con la Fig. 1, y eliminando los subíndices  $i$  de cada canal para simplificar la notación, obtenemos:

$$x(t) \approx z(t) = g(e(t)) = \sum_{n=1}^{N+1} a_n e(t)^{n-1} \quad (4)$$

Necesitamos calcular la entropía  $H(Z)$  de cada señal  $z$ .

Esta entropía se escribe como:

$$H(z) = H(e) + E \left[ \sum_i \log g'(e_i) \right] \quad (5)$$

Y las derivadas de esta ecuación con respecto a los parámetros  $a_p$  (coeficientes del polinomio) son:

$$\begin{aligned} \frac{\partial H(z)}{\partial a_p} &= E \left[ \left( \sum_{n=1}^{N+1} a_n (n-1) e(\tau)^{n-2} \right)^{-1} \frac{\partial}{\partial a_p} g'(e(\tau)) \right] \\ &= E \left[ \left( \sum_{n=1}^{N+1} a_n (n-1) e(\tau)^{n-2} \right)^{-1} (p-1) e(\tau)^{p-2} \right] \end{aligned} \quad (6)$$

Finalmente, proponemos un algoritmo de gradiente para estimar la función inversa  $g$  que queda como:

$$a \leftarrow a + \mu \Delta a \quad (7)$$

### III. RESULTADOS EXPERIMENTALES

Con el fin de poder comparar los resultados obtenidos con el nuevo método propuesto, vamos a realizar los experimentos con los tres métodos siguientes: Gaussianización [1], uniformización (similar a [1] pero forzando distribución uniforme para  $z$ ) y el algoritmo propuesto (maximización de entropía).

Se darán solamente los resultados de la inversión de la parte no lineal. La parte lineal se podría invertir con cualquier método de separación de mezclas lineales existente.

En todos los experimentos fijamos los siguientes parámetros:

- Matriz de mezcla:  $A = \begin{bmatrix} 1 & 0.6 \\ 0.7 & 1 \end{bmatrix}$
- Funciones no lineales:  $f_1 = f_2 = \tanh(3x) + 0.1x$
- Longitud de las señales:  $T = 500$  muestras
- Grado de los polinomios para las estimaciones de  $g_1$  y  $g_2$ :  $N = 10$
- Numero máximo de iteraciones para la estimación de  $g_1$  y  $g_2$ :  $iter = 100$
- Ganancia de adaptación:  $\mu = 0.01$

La ganancia en la inversión de las funciones no lineales obtenida con los tres métodos la vamos a calcular mediante:

$$SNR_i(dB) = 10 \log \left( \frac{z_i^2}{z_i^2 - x_i^2} \right) \quad (9)$$

Con  $i=1,2$ .

En el primer experimento utilizamos dos ruidos aleatorios como señales fuente, generadas mediante la función de Matlab `rand([1 T]) - 0.5`. La Fig. 2 muestra la evolución de la SNR en dB obtenida con los tres métodos.

Podemos observar que el algoritmo propuesto obtiene mejores resultados que la Gaussianización o la uniformización. Además, obtenemos mejores resultados con la Gaussianización que con la uniformización. Esto es debido a que las fuentes son señales de ruido continuas. Por otra parte, aunque el método propuesto es un método iterativo, la convergencia es muy rápida, estabilizando el resultado en menos de 50 iteraciones.

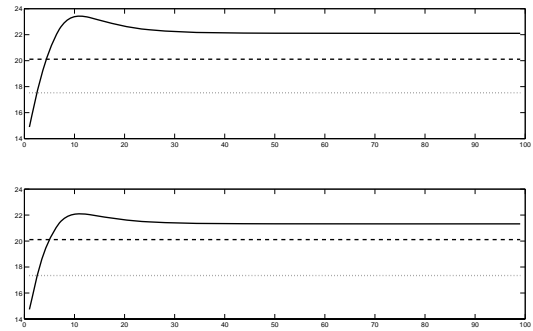


Fig. 2: SNR (dB) de la compensación de la función no lineal para dos fuentes de ruido (continuo) aleatorio distribuido uniformemente. En la parte superior, los resultados para el canal 1, y en la parte inferior los resultados para el canal 2. El trazo sólido corresponde al método de maximización de entropía, el trazo discontinuo corresponde al método de Gaussianización, y el trazo de puntos corresponde al método de uniformización.

En el segundo experimento utilizamos una fuente de ruido aleatorio y otra determinista, generada mediante el siguiente código Matlab: `sin((1:T)/sqrt(2)) + sin((1:T)/sqrt(3))`. La Fig. 3 muestra evolución de la SNR en dB obtenida con los tres métodos. Una vez más, el método propuesto obtiene mejores resultados que los otros dos, pero en este caso el método de uniformización gana al de Gaussianización, dada la naturaleza determinista de una de las fuentes.

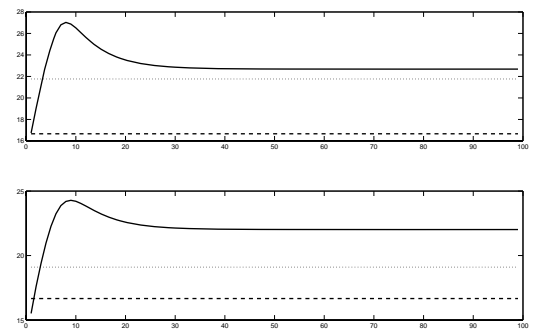


Fig. 3: SNR (dB) de la compensación de la función no lineal para dos fuentes continuas (un ruido y una señal periódica). En la parte superior, los resultados para el canal 1, y en la parte inferior los resultados para el canal 2. El trazo sólido corresponde al método de maximización de entropía, el trazo discontinuo corresponde al método de Gaussianización, y el trazo de puntos corresponde al método de uniformización.

Finalmente, en la Fig. 4 se muestran los resultados obtenidos con dos ruidos uniformes discretos de 5 niveles  $(-0.4, -0.2, 0, +0.2, +0.4)$ . En este caso, dado que las dos fuentes son discretas y tienen una distribución uniforme, el método de uniformización supera al de Gaussianización,

pero el método propuesto nos da una vez más el mejor resultado entre los tres.

Para ilustrar mejor la calidad de la inversión conseguida con este método, en la Fig. 5 se muestra un ejemplo (ideal) de funciones  $f$ ,  $g$  y  $(g \circ f)$ . Podemos compararlo con el resultado (señales del primer ejemplo) de  $(g \circ f)$  obtenido mediante los tres métodos, mostrados en la figura Fig. 6. En la izquierda se muestra el resultado obtenido mediante maximización de entropía, en el centro el resultado obtenido mediante uniformización, y en la derecha el resultado obtenido mediante Gaussianización. Se puede observar que el nuevo método obtiene un resultado mejor que los otros dos, dando una función  $(g \circ f)$  casi lineal. La compensación es muy buena en todo el rango de valores, y especialmente mucho mejor que los otros dos métodos para los valores grandes de la distribución.

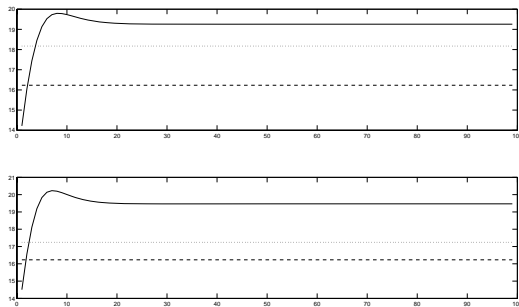


Fig. 4: SNR (dB) de la compensación de la función no lineal para dos fuentes de ruido (discreto) aleatorio distribuido uniformemente. En la parte superior, los resultados para el canal 1, y en la parte inferior los resultados para el canal 2. El trazo sólido corresponde al método de maximización de entropía, el trazo discontinuo corresponde al método de Gaussianización, y el trazo de puntos corresponde al método de uniformización

#### IV. CONCLUSIONES

En este artículo proponemos una generalización del método de Gaussianización para la inversión ciega de funciones no lineales en mezclas PNL.

El método se basa en la maximización de la entropía de las salidas no lineales para aproximar las funciones no lineales desconocidas  $g$ . Dado que la entropía máxima (para una varianza fijada) se obtiene para una distribución Gaussiana en el caso de variables aleatorias continuas, y para una distribución uniforme en el caso de variables aleatorias discretas, proponemos invertir la parte no lineal mediante la maximización de la entropía.

Los resultados experimentales muestran que el método propuesto da buenos resultados y consigue linealizar las observaciones. De hecho, el método propuesto permite desacoplar la estimación de la parte no lineal de la estimación de la parte lineal (inversión de la matriz de mezcla). Una vez resuelta la inversión de la parte no lineal, cualquier algoritmo existente para la separación de mezclas lineales puede ser utilizado para invertir la matriz de mezcla.

El uso de polinomios para la parametrización de las funciones desconocidas  $g$  nos da un algoritmo con pocos parámetros. De todas formas, se debe investigar otras posibilidades dado que un polinomio no sería útil en el caso de tener que invertir funciones con cambios bruscos. En este caso, se podría optar por una red neuronal o una aproximación con *splines*.

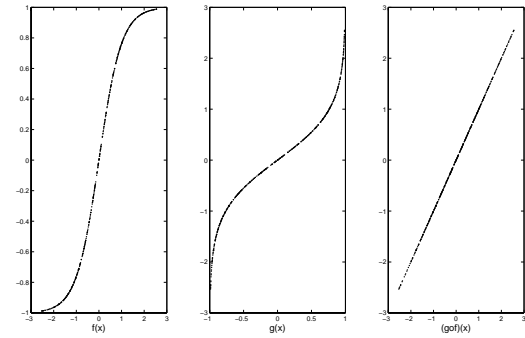


Fig. 5: Izquierda: función no lineal  $f(x)$ . Centro: función inversa (teórica)  $g(x) = f(x)^{-1}$ . Derecha: composición de las dos funciones  $(g \circ f)(x)$ , dando como resultado una recta.

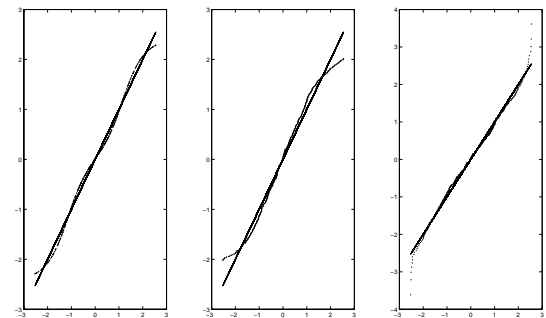


Fig. 6: Composición de las dos funciones no lineales  $(g \circ f)(x)$  para los tres métodos. Izquierda: maximización de entropía. Centro: uniformización. Derecha: Gaussianización. Para poder comparar los tres métodos, se superpone la función identidad (caso ideal) a las tres gráficas.

Finalmente, y como mostraremos en futuros trabajos, si una mezcla PNL se puede desacoplar en dos partes (no lineal y lineal) que se pueden invertir de forma separada, este método se puede utilizar también para mezclas convolutivas PNL. En este caso, una vez invertidas las funciones no lineales con nuestro método, se resuelve la parte lineal (inversión de la matriz de filtros) con cualquier de los métodos existentes de inversión de mezclas convolutivas. Los resultados experimentales obtenidos son muy satisfactorios y confirman la validez del método.

#### AGRADECIMIENTOS

Este trabajo se ha realizado gracias al proyecto R0912 de la Universitat de Vic.

#### REFERENCIAS

- [1] J. Solé-Casals, C. Jutten, D. T. Pham, "Fast approximation of nonlinearities for improving inversion algorithms of PNL mixtures and Wiener systems", *Signal Processing* vol. 85 (2005) pp. 1780-1786.
- [2] T.M. Cover, J.A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, 1991.
- [3] A. Taleb, J. Solé-Casals, C. Jutten, "Quasi-Nonparametric Blind Inversion of Wiener Systems" *IEEE Transactions on Signal Processing*, vol. 49 (2001), pp. 917-924.
- [4] A. Ziehe, M. Kawanabe, S. Harmeling, K.-R. Müller, "Blind Separation of Post-Nonlinear Mixtures using Gaussianizing Transformations and Temporal Decorrelation", *Proceedings of the 4<sup>th</sup> International Symposium on Independent Component Analysis (ICA2003)*, Nara (Japan), April 2003, pp. 269-274.