Initialisation of Nonlinearities for PNL and Wiener systems Inversion

* Jordi SOLE-CASALS[1,2], Christian JUTTEN[2], and Dinh-Tuan PHAM[3]

[1] Signal Processing Group, Universitat de Vic, Sagrada Família 7, 08500, Vic, Spain
jordi.sole@uvic.es
[2] Laboratoire des Images et des Signaux (CNRS UMR $n°5083$)
INPG, 46 Avénue Félix Viallet-38031 Grenoble Cedex, France
Christian.Jutten@inpg.fr
[3] Laboratoire de Modélisation et de Calcul, BP 53, 38041 Grenoble Cedex 9, France
Dinh-Tuan.Pham@imag.fr

Abstract. This paper proposes a very fast method for blindly initializing a nonlinear mapping which transforms a sum of random variables. The method provides a surprisingly good approximation even when the basic assumption is not fully satisfied. The method can been used successfully for initializing nonlinearity in post-nonlinear mixtures or in Wiener system inversion, for improving algorithm speed and convergence.

1 Introduction

Blind Separation of independent sources (BSS) is a basic problem in signal processing, which has been considered intensively in the last fifteen years, mainly for linear (instantaneous as well as convolutive) mixtures. More recently, a few researchers [1–10] addressed the problem of source separation in nonlinear mixtures, whose observations are $\mathbf{e} = f(\mathbf{s})$. Especially Taleb and Jutten [8] have studied a special and realistic case of nonlinear mixtures, called post nonlinear (PNL) mixtures which are separable. As shown in Figure 1, this two-stage system consists of a linear mixing matrix, followed by componentwise nonlinear distortions. It then provides the mixing observations:

$$e_i(t) = f_i(\sum_j a_{ij} s_j(t)) \tag{1}$$

where $s_j(t)$ are the independent sources, $e_i(t)$ is the i-th observation, a_{ij} denotes the entries of the unknown mixing matrix $\mathbf{A}$, and f_i is the unknown nonlinear mapping on the component i.
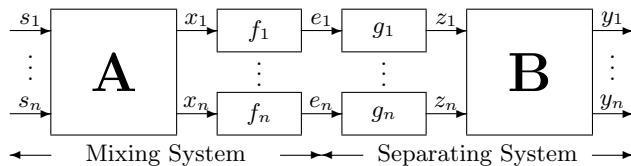

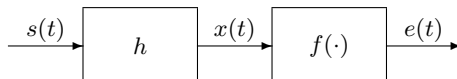**Fig. 1.** The mixing-separating system for PNL mixtures.

Fig. 2. A Wiener system consists of a filter followed by a distortion

The problem of blind inversion of Wiener systems (Fig. 2) is equivalent to the source separation problem in PNL mixtures [11]. Its output writes as

$$e(t) = f(\sum_k h(k)s(t-k)) \tag{2}$$

where $s(t)$ is the independent and identically distributed (iid) input, $e(t)$ is the observation, $h(k)$ denotes the entries of the unknown filter H and f is the unknown nonlinear mapping, assumed invertible and memoryless.

Blind separation or inversion of the above models requires first to estimate the inverse of the nonlinear mapping, and then to inverse the linear part. This can be done by minimizing the mutual information of the inversion structure output. However, it leads to slow and cost consuming algorithms, since the two parts are in cascade and optimized with the same criterion.

In this paper, we propose a simple, very fast and efficient method for roughly estimating the inverse of the nonlinear mapping. Section 2 explains the principles. Section 3 proposes two classes of algorithms and section 4 shows some experimental results where one use this method for initializing blind inversion algorithms and improving its speed.

2   Principles

2.1   The basic assumption

In the model (1), consider the signal just before the nonlinear mapping. For the i-th component in the PNL mixture, The signal $x_i(t) = \sum_j a_{ij}s_j(t)$ is a weighted sum of random variables. According to the Central Limit Theorem, X_i tends toward a Gaussian random variable. The nonlinear mapping f_i changes the distribution, and consequently we can assume that the random variable
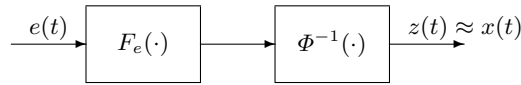
Fig. 3. The inversion system

$E_i = f_i(X_i)$ is far from Gaussian. Then, we propose to estimate the inverse of f_i, as the nonlinear mapping g_i which enforces the random variable $Z_i = g_i(E_i)$ to be Gaussian.

Similarly, in the Wiener systems the filtered signal $x(t) = \sum_k h(k)s(t-k)$, just before the nonlinearity, is a weighted sum of random variables. According to the Central Limit Theorem, the random variable X, associated to $x(t)$, tends to be a Gaussian random variable. Of course, the vicinity to a Gaussian variable depends on the filter, but X is closer to a Gaussian distribution than the distribution of the original source S. We then propose to approximate the inverse of $f(\cdot)$ by the function $g(\cdot)$ such that $g(E)$ is Gaussian.

In the next section, since the two problems are very similar, we cancel the index i for simplifying the notations.

2.2   Cumulative density function

The simplest approach for computing g_i is based on the property of the cumulative density function. Consider the random variable E, and denote $F_E(u)$ its cumulative density function:

$$F_E(u) = Pr(E < u) \tag{3}$$

where $Pr()$ denotes the probability.

The random variable $U = F_E(E)$ is then uniformly distributed in $[0, 1]$. Denoting $\Phi(u)$ the Gaussian cumulative density function, which transforms a unit variance Gaussian variable in a uniform random variable in $[0, 1]$, it is clear that $\Phi^{-1}(U)$ is a unit variance Gaussian random variable.

Then, a simple approximation of the inverse g of the nonlinear mapping f is:

$$\hat{g} = \Phi^{-1} \circ F_E \tag{4}$$

2.3   Maximization of Shannon entropy

Consider now the Shannon entropy of the unit variance random variable Z

$$H(Z) = \int -log p_Z(u) p_Z(u) du \tag{5}$$

where $p_Z(u)$ denotes the probability density function.

Since, for unit variance random variable, the Shannon entropy $H(Z)$ is maximum if Z is Gaussian [12], then g can be estimated so that $H(Z) = H(g(E))$ is maximum (under the constrain of unit variance).

3 Algorithms

Using the previous results, one can propose two simple algorithms for the rough estimation of the inverse of the nonlinear mapping f. The first algorithm (see Fig. 3) is based on the formula (4) derived in Subsection 2.2. The Matlab code is very simple and very fast.

A second algorithm, based on the result on Subsection 2.3, consists in adjusting a nonlinear mapping g so that the Shannon's entropy of z is maximum under the constraint $Ez^2 = 1$ (see [10] for a similar work). We can parametrize the nonlinear function g, for example by means of neural networks (multylayer perceptron), as showed in Fig. 4.
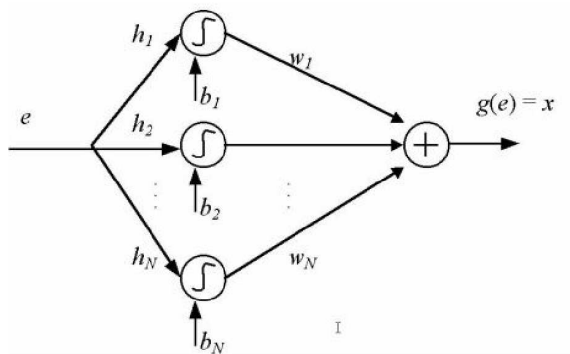

Fig. 4. Multilayer perceptron

A gradient approach can be easily derived for adjusting vector parameters $[\mathbf{h}, \mathbf{b}, \mathbf{w}]$, in order to optimize the cost function $J = max_{\mathbf{h}, \mathbf{b}, \mathbf{w}}(H(Z))$. Although the second idea is still quite simple, it leads to an algorithm which is much more complicated and longer to converge than the previous one.

Hence, in the following, we only give experimental results with the simplest and more efficient algorithm based on (4).

4 Experimental results

4.1 Protocol

In order to test the algorithm, we made experiments, using source signals with different kurtosis.

Mixing of variables can be done according to either linear mixtures $\mathbf{A}$ or linear filtering h. For simplicity, we only report results with linear filtering according to (2), but similar results are obtained with (1) [13].

Ten different filters (low-pass, high-pass or band-pass) have been used for providing filtered signals with various distributions. Then, we apply the non-linearity f. We check, as expected by theory, that the algorithm is completely independent of f since, $\forall f$ the function $\Phi^{-1} \circ F_E \circ f$ transforms the random variable X to a Gaussian variable Z. If the compensation of the nonlinearity was perfect, the function $\Phi^{-1} \circ F_E \circ f$ should be the identity function. Of course, it can be rigorously true, only if X is Gaussian.

For testing the algorithm of PNL source separation, we did some experiments for mixtures of two uniformly distributed random sources. The method used is the algorithm proposed by Taleb and Jutten [8], denoted TJ. The mixing system is composed of:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.4 \\ 0.7 & 1 \end{bmatrix} \tag{6}$$

$$f_1(x) = f_2(x) = 0.1x + \tanh(10x) \tag{7}$$

This mixture leads to the joint distribution showed in Fig. 5, where the effect of the nonlinearities is clearly visible (left). In the same figure we can see the scatter plot after the initialization of nonlinear functions g (center) and the scatter plot of the initialized outputs $\mathbf{y}$, where the signals are decorrelated (right). It is easy to see qualitatively the initialization provides an estimation $y(t)$ which is a mixture, simpler than $e(t)$.
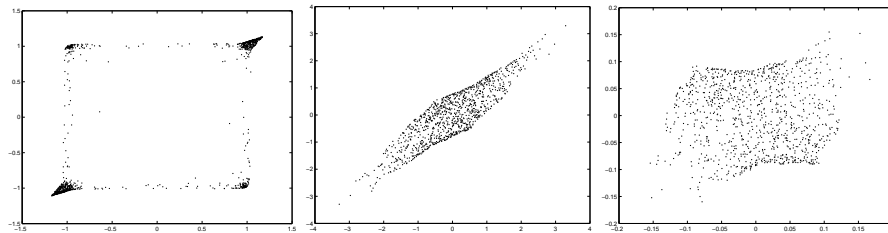

Fig. 5. Scatter plot of the observed signals (left), of the signals after initializing non-linear functions g (center) and of the decorrelated output signals (right).

Despite hard nonlinearities $(0.05x + tanh(10x))$ are used in the experiments, the results obtained with the TJ method are satisfactory. The initialization process increases the convergence speed of the algorithm, and sometimes gives a better result in terms of output SNR (Fig. 6).

4.2 Results

The performance index ε of the compensation will be simply the mean square error[4]:

$$\varepsilon = E_X[\|(\Phi^{-1} \circ F_E \circ f)(X) - X\|^2] \tag{8}$$

---

[4] For computing ε, $x(t)$ and $z(t) = (\Phi^{-1} \circ F_E \circ f)(x(t))$ have to be normalized
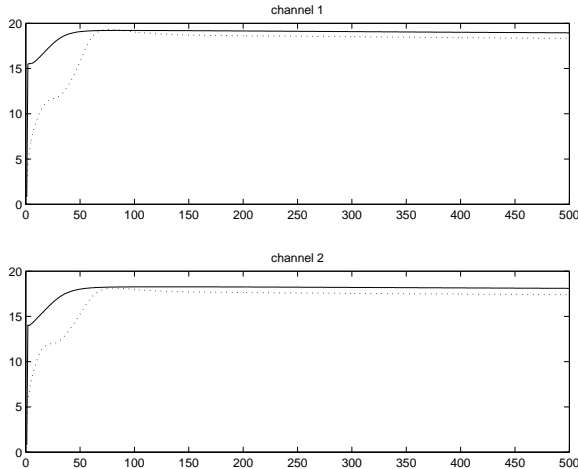
Fig. 6. SNR evolution for TJ algorithm with initialization process(solid line) and without (dashed line).

which measures the divergence between $\Phi^{-1} \circ F_E \circ f$ and the linear function x.

Fig. 7 shows the performance index versus the kurtosis of the filtered signals X, obtained with the different sources and filters. One can remark that the performance is maximum when the signal kurtosis is close to zero, *i.e.* X is close to a Gaussian, and vanishes as the kurtosis moves away from zero.

As a result, the efficacy of the method is only related on the distribution of X, just before the nonlinearity f: closer to the Gaussian the distribution, closer to zero the error ε.

In Fig. 8 and 9 we show an example of good and poor compensation of the nonlinear function. The good compensation corresponds to 0.0238 kurtosis and the poor case to -1.2631.

5   Conclusion

In this paper, we propose a very simple and fast method for blindly approximating nonlinear mapping. The method is based on the assumption that the input variable of the nonlinear mapping is Gaussian due to mixture or filtering. The results show the method is robust enough to the Gaussian assumption. In the worst case (see Fig. 9), it leads to a rough approximation of the nonlinear mapping which can be used as initialization value for enhancing BSS or Wiener algorithms.

References

1. G. Burel, "Blind separation of sources: a nonlinear neural algorithm," *Neural Networks*, vol. 5, pp. 937–947, 1992.
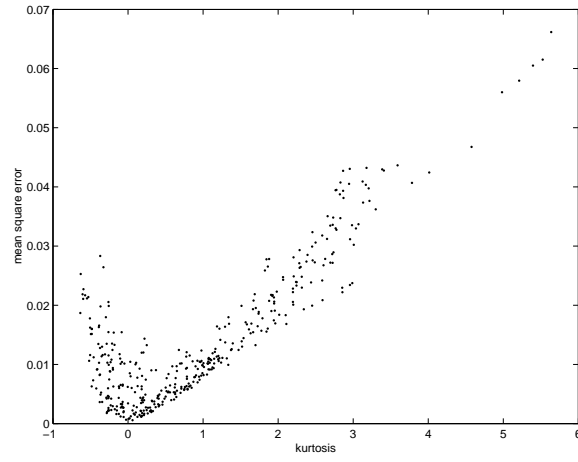
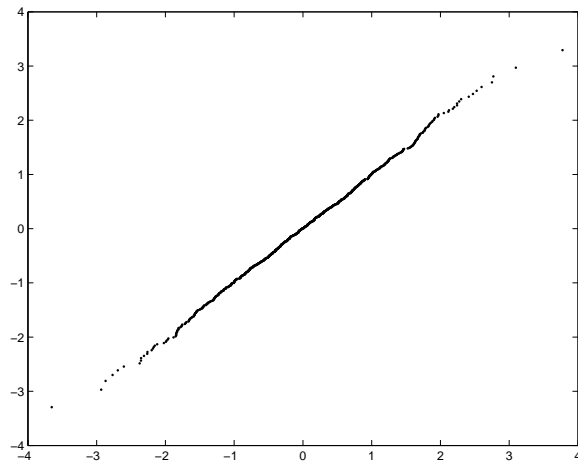**Fig. 7.** Performance index ε versus kurtosis

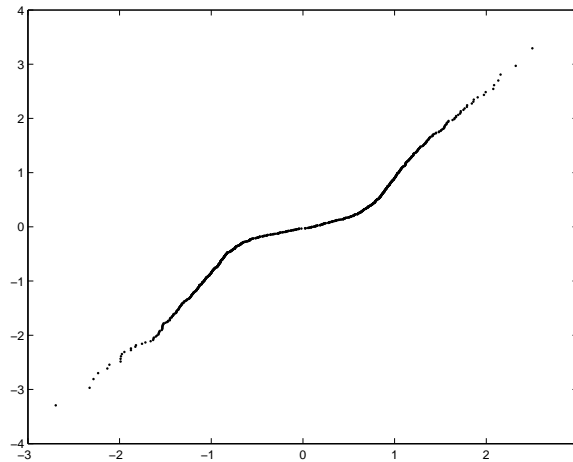**Fig. 8.** Best case nonlinear function compensation

Fig. 9. Worst case nonlinear function compensation

2. L. Parra, G. Deco, and S. Miesbach, "Statistical independance and novelty detection with information preserving nonlinear maps," *Neural Computation*, vol. 8, pp. 260–269, 1996.
3. P. Pajunen, A. Hyvärinen, and J. Karhunen, "Non linear source separation by self-organizing maps," in *ICONIP 96*, Hong-Kong, September 1996.
4. A. Taleb and C. Jutten, "Entropy optimization, application to blind source separation," in *ICANN 97*, Lausanne (Switzerland), October 1997, pp. 529–534.
5. T.-W. Lee, B. Koehler, and R. Orglmeister, "Blind separation of nonlinear mixing models," in *IEEE NNSP*, Florida, USA, 1997, pp. 406–415.
6. C.G. Puntonet, M.R. Alvarez, and B. Prieto, A. Prieto, "Separation of sources in a class of post-nonlinear mixtures," in *ESANN 98*, Bruges, Belgium, April 1998.
7. H. H. Yang, S. Amari, and A. Cichocki, "Information-theoritic approach to blind separation of sources in nonlinear mixture," *Signal Processing*, vol. 64, no. 3, pp. 291–300, 1998.
8. A. Taleb and C. Jutten, "Source separation in post nonlinear mixtures," *IEEE Tr. on Signal Processing*, vol. 47, no. 10, pp. 2807–2820, 1999.
9. G. C. Marques and L. B. Almeida, "Separation of nonlinear mixtures using pattern repulsion," in *ICA 99*, Aussois (FRANCE), 1999, pp. 277–282.
10. A. Ziehe, M. Kawanabe, S. Harmeling, and K.R. Muller, "Separation of post-nonlinear mixtures using ace and temporal decorrelations," in *ICA 2001*, San Diego - California, 2001, pp. 433–438.
11. A. Taleb, J. Solé-Casals, and C. Jutten, "Quasi-nonparametric blind inversion of Wiener systems," *IEEE Tr. on Signal Processing*, vol. 49, no. 5, pp. 917–924, 2001.
12. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, 1991.
13. J. Solé-Casals, M. Babaie-Zadeh, C. Jutten, and D.T. Pham, "Improving algorithm speed in PNL mixture separation and Wiener system inversion," in *Proceedings of ICA'03*, Nara, Japan, 2003.