RESEARCH ARTICLE

Inverting Monotonic Nonlinearities by Entropy Maximization

Jordi Solé-Casals[1]*, Karmele López-de-Ipiña Pena[2], Cesar F. Caiafa[3]

1 Data and Signal Processing Research Group, U Science Tech, University of Vic – Central University of Catalonia, Vic, Catalonia, Spain, **2** Systems Engineering and Automation Department, Universidad del País Vasco/Euskal Herriko Unibertsitatea, EleKin Research Group, Donostia, Spain, **3** Instituto Argentino de Radioastronomía (IAR), CONICET, CCT – La Plata, Buenos Aires, Argentina

* jordi.sole@uvic.cat

OPEN ACCESS

Citation: Solé-Casals J, López-de-Ipiña Pena K, Caiafa CF (2016) Inverting Monotonic Nonlinearities by Entropy Maximization. PLoS ONE 11(10): e0165288. doi:10.1371/journal.pone.0165288

Editor: Kay Hamacher, Technische Universitat Darmstadt, GERMANY

Received: December 4, 2015

Accepted: October 10, 2016

Published: October 25, 2016

Data Availability Statement: All files are available from the GitHub public repository (https://github.com/Sole-Casals/MaxEnt).

Funding: This work has been partly funded by the University of Vic – Central University of Catalonia under the grant R0947 to JSC and grants from ANPCyT - PICT 2012 -1519 and CONICET - PIP 114-201101-00021, Argentina, to CFC. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

This paper proposes a new method for blind inversion of a monotonic nonlinear map applied to a sum of random variables. Such kinds of mixtures of random variables are found in source separation and Wiener system inversion problems, for example. The importance of our proposed method is based on the fact that it permits to decouple the estimation of the nonlinear part (nonlinear compensation) from the estimation of the linear one (source separation matrix or deconvolution filter), which can be solved by applying any convenient linear algorithm. Our new nonlinear compensation algorithm, the MaxEnt algorithm, generalizes the idea of Gaussianization of the observation by maximizing its entropy instead. We developed two versions of our algorithm based either in a polynomial or a neural network parameterization of the nonlinear function. We provide a sufficient condition on the nonlinear function and the probability distribution that gives a guarantee for the MaxEnt method to succeed compensating the distortion. Through an extensive set of simulations, MaxEnt is compared with existing algorithms for blind approximation of nonlinear maps. Experiments show that MaxEnt is able to successfully compensate monotonic distortions outperforming other methods in terms of the obtained Signal to Noise Ratio in many important cases, for example when the number of variables in a mixture is small. Besides its ability for compensating nonlinearities, MaxEnt is very robust, i.e. showing small variability in the results.

1. Introduction

Nonlinear models are powerful tools for modelling practical situations when linear models fail. This is the case of post-nonlinear (PNL) source separation problems and nonlinear blind deconvolution scenarios. In real world situations, usually we do not have access to the distortion input. Hence we cannot use traditional methods, which assume that both the input and the output of the distortion are available [1]. Examples of such traditional methods are those based on higher-order input/output cross-correlation [2], bispectrum estimation [3, 4] or on the application of the Bussgang and Prices theorems [5, 6] for nonlinear systems with Gaussian inputs. In this work, we will focus only on blind identification methods.

Competing Interests: The authors have declared that no competing interests exist.

Blind source separation with PNL mixtures (see Fig 1a), requires estimating the inverses of the nonlinear maps f_n and the inverse of the mixing matrix **A**(linear part). This can be done by minimizing the Mutual Information (MI) of the inversion structure output, i.e. by solving an optimization problem over the parameters of the linear and nonlinear parts together, as detailed in [7]. However, this leads to complex and slow algorithms.

On the other hand, in a single channel scenario, there is a particular class of nonlinear systems composed by a linear filter followed by a memoryless nonlinear distortion (Fig 1b). This class of nonlinear systems, also known as a Wiener system, is not only a nice and mathematically attracting model, but also a realistic model used in various areas, such as biology [8], industry [9], sociology and psychology (see also [10] and the references therein). A fully blind procedure for finding the inverse system (Hammerstein system), when the (unknown) input signal $s(t)$ is a non-Gaussian i.i.d. process, is detailed in [11, 12]. If the input process is not i.i.d. but is a linear filtered version of an i.i.d. noise (the so-called innovation), the output $y(t)$ provides the innovation instead of the input signal. The restitution of the input then requires the prior knowledge (or the estimation) of the filter generating the signal $s(t)$, as detailed in [12].

A simple and very fast method for roughly estimating the inverse of a nonlinear map has been proposed in [11] and used in [12], which will be referenced here as the *Gaussianization* algorithm (see **Section 2.1**). This method is very attractive because it focuses only on the nonlinear compensation estimation. After the nonlinear compensation is found, we can solve the linear part (inverse matrix or deconvolution filter) by applying any available linear algorithm. This nonlinear-linear decoupling approach allows one to obtain simpler and faster algorithms compared to global algorithms that estimate the linear and nonlinear parts together [7,11,12].

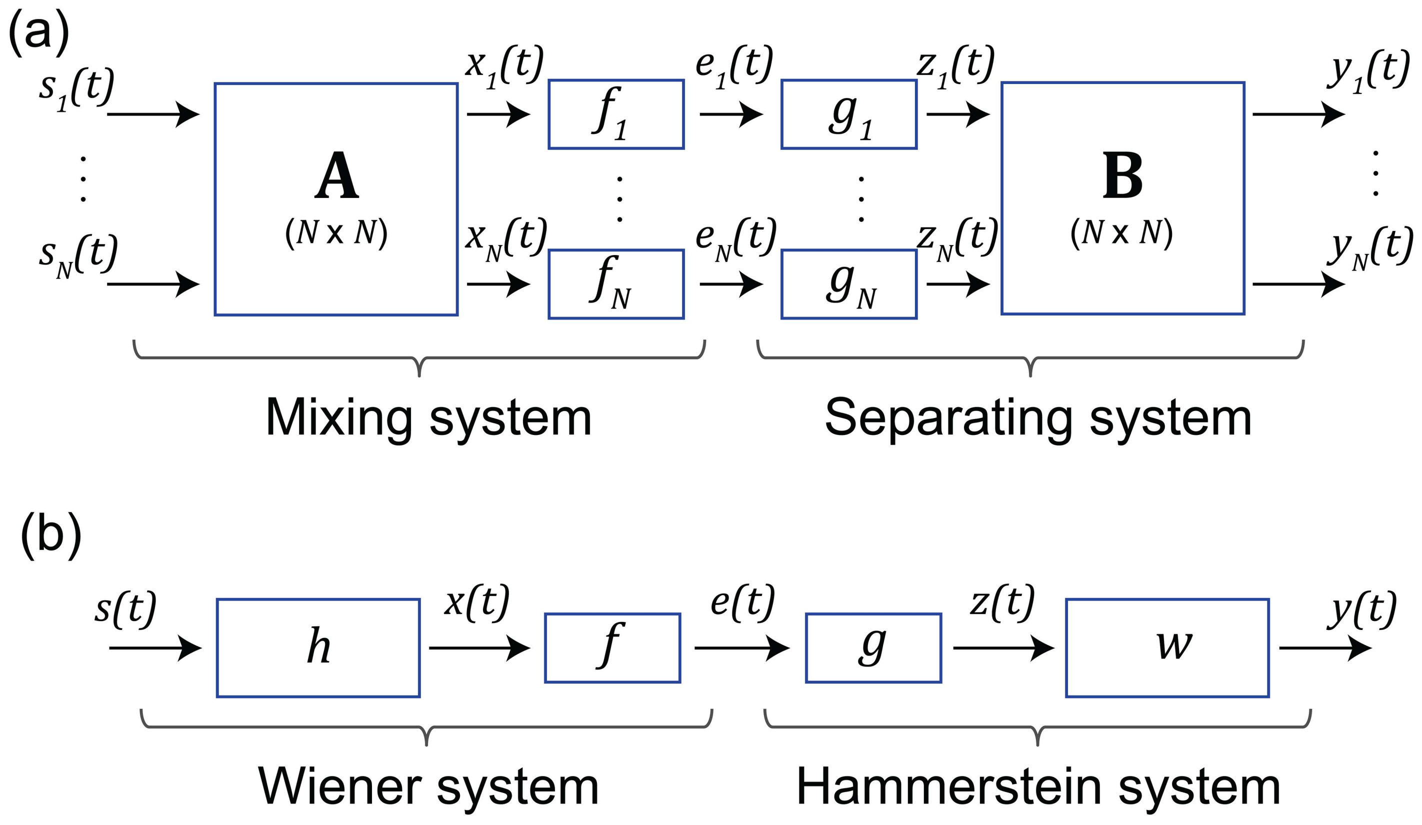

Fig 1. Typical applications where nonlinear compensation is required: (a) the mixing—separating system for the source separation problem with PNL mixtures and (b) the Wiener-Hammerstein system for deconvolution problem with PNL distortion.

doi:10.1371/journal.pone.0165288.g001

The basic idea for the Gaussianization algorithm is to notice that signals $x_n(t)$ in Fig 1a and $x(t)$ in Fig 1b, i.e. just before the nonlinear map, are a weighted sum of random variables, which in the case of variables with finite second order moments, is more Gaussian than the individual inputs. In fact, according to the Central Limit Theorem, the random variable X_n tends toward a Gaussian variable as $N \to \infty$ (for finite second order moments). Note that we use capital letters to refer to a random variable associated to a signal. For example, a signal $x(t)$ is considered as a time series (realizations) obtained from a random variable X. The nonlinear mapping function f_n changes the distribution, and consequently we can assume that the random variable $E_n = f_n(X_n)$ is more distant from a Gaussian than X_n. Then, the algorithm estimates the inverse of f_n as the nonlinear mapping function g_n which enforces the random variable $Z_n = g_n(E_n)$ to be exactly Gaussian.

In this paper we will focus only on the nonlinear compensation, assuming a monotonic (unknown) distortion, generalizing the idea of Gaussianization for estimating the inverse of the nonlinear map by using maximization of entropy strategy instead. We provide a sufficient condition under which the output of a nonlinear transform has smaller entropy compared with its input, thus giving a guarantee for the MaxEnt method to succeed inverting the nonlinear distortion. The new algorithm consists in adjusting a nonlinear mapping function g_n, through an appropriate parameterization, so that the entropy of $Z_n = g_n(E_n)$ is maximum under the constraint of unit variance. Although the idea is similar to that used in [13], it differs since the recovered random variable z (after nonlinear inversion) has not necessarily a Gaussian distribution, but maximum entropy. In **Section 2** we explain the theoretical principles of the Gaussianization and Uniformization algorithms and introduce the equations for our new proposed method: the MaxEnt algorithm. In **Section 3**, we present extensive experimental results showing the robustness and performance of our new method compared to the Gaussianization and Uniformization methods and we compare MaxEnt against a state-of-the-art global optimization based method. In **Section 4**, our main conclusions are outlined.

2. Materials and Methods

In this article we denote random variables by capital letters and their realizations by the corresponding lower case letter. For example, X and x, correspond to a random variable and its realization (value), respectively.

Under the assumption that a random variable X is close to a Gaussian, if we let $E = f(X)$, then we may estimate the inverse $g = f^{-1}$ by enforcing $Z = g(E)$ to be Gaussian [13]. We can generalize this idea by observing that entropy at the output of the linear system X is always higher than the entropy of each independent input S_n, under the constant variance constraint (see Fig 1a). Additionally, if we assume that the nonlinear distortion has the effect to decrease the entropy, then we can estimate the inverse $g = f^{-1}$ searching, over the space of plausible nonlinear maps, the one that makes the entropy of $Z = g(E)$ to be maximum under the unit-variance constraint. We call this new method as the MaxEnt nonlinear compensation method and it is described in detail in **Section 2.2**.

It is well known that the maximum entropy distribution for unbounded support under mean and variance constraints is the Gaussian distribution, whereas for bounded supports under no (but adding-up) constraint is the uniform distribution. However, it is important to note that, by using a Maximum Entropy algorithm (MaxEnt) we are not enforcing directly to obtain a Gaussian or uniform distribution for Z. However, our extensive experimental results (see Section 3) confirm the fact that for most important cases, MaxEnt outperforms Gaussianization and Uniformization strategies and gives also more robust results.

In the following sections, the theoretical foundations are presented and the algorithms are derived for the Gaussianization, Uniformization and MaxEnt methods.

2.1. Gaussianization and Uniformization

The simplest approach for computing the inverse system g by Gaussianization [13] is based on the property of the cumulative density function (cdf). Consider the random variable E, and denote its cdf $F_E(u) = \mathrm{P_r}(E < u)$ where $\mathrm{P_r}$ denotes the probability. The random variable $U = F_E(E)$ is then uniformly distributed in [0, 1]. Denoting by $\Phi(u)$ the Gaussian cdf, which transforms a unit variance Gaussian variable into 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 function f is $\hat{g} = \Phi^{-1} \circ F_E$ [13,14].

In the case of variables with bounded support and no constraints, the maximum entropy is achieved by the uniform distribution. Therefore, we also consider Uniformization as a possible way to compute the inverse g. The same strategy explained for Gaussianization, but using the cdf for the uniform distribution is then used for computing the monotonous nonlinear mapping g.

2.2. Entropy maximization (MaxEnt)

We assume that the nonlinear transformation $E = f(X)$, has the effect to decrease the normalized (unit-norm) entropy of the input, i.e. $H(E) < H(X)$, so the transform can be blindly estimated by MaxEnt. We know this is true when the input variable X is of maximum entropy, e.g. Gaussian, but it is not clear a priori, under which conditions on the distribution of X and the nonlinearity $f(\cdot)$ the MaxEnt method is well posed. The following theorem states a sufficient condition for the decrease of entropy under a nonlinear transformation. As we demonstrate in **Section 3.7**, through experimental validation, this condition holds true in most practical cases.

Theorem 1 (sufficient condition for entropy decrease under a nonlinear transform): *Given a zero-mean and unit-variance variable X and a monotonic and antisymmetric nonlinear function $f(\cdot)$, if the following condition holds true*:

$$\mathrm{E}^2[f'(\mathrm{x})] < \mathrm{E}[f^2(\mathrm{x})] \tag{1}$$

then the entropy of the normalized output variable $E = f(X)$ is smaller than the input variable X.

***Proof*: see** Appendix

In order to understand the consequences of this result, the following corollary, provides the theoretical justification of the MaxEnt method for a particular family of nonlinearities, found in many practical applications [15].

Corollary 1:(particular case $f(x) = x^3 + \beta x$ *with* $\beta \geq 0$):

For any probability density function (pdf) $p_X(x)$ *and the parameterized nonlinear function* $f(x) = x^3 + \beta x$, *the entropy of the normalized output variable* $E = f(X)$ *is smaller than the input variable X in the following cases*:

1. *Super-Gaussian or zero excess kurtosis* ($\mu_4 \geq 3$), $\mu_6 > 9$ *and any* $\beta \geq 0$;
2. *Super-Gaussian* ($\mu_4 > 3$), $\mu_6 < 9$ *and* $\beta > 0.5(9 - \mu_6)/(\mu_4 - 3)$;
3. *Sub-Gaussian* ($\mu_4 < 3$), $\mu_6 > 9$ *and* $\beta < 0.5(\mu_6 - 9)/(3 - \mu_4)$;

where μ_p *is the moment of order p. It is noted that, in our case with standardized variables* ($\mathrm{E}(X) = 0$ *and* $\mathrm{E}[X^2]$), μ_4 *and* μ_6 *correspond to the kurtosis and the hyper-flatness statistical measures, respectively*.

***Proof*: see** Appendix

It is important to highlight that this corollary gives us a guarantee for the MaxEnt method to work for the case of for super-Gaussian or zero excess kurtosis variables when $\mu_6 > 9$ and give us clear conditions on the sixth order moment μ_6 and parameter β for the case of sub-Gaussian variables.

2.2.1. Derivation of the MaxEnt algorithm. Consider the entropy of the continuous unit variance random variable $Z = g(E)$:

$$H(Z) = -\mathrm{E}[\log(p_Z(z))] \tag{2}$$

where $p_Z(z)$ denotes the pdf of random variable $Z = g(E)$. By using a well-known basic property of the entropy, $H(Z)$ can be written as follows:

$$H(Z) = H(E) + \mathrm{E}[\log|g'(e)|] \tag{3}$$

We can consider different ways to parameterize the nonlinearity $g(e)$ and maximize eq (3) in terms of the used parameterization. In the following sections, two different parameterizations are proposed: polynomial and neural network parameterizations.

2.2.2. Polynomial parameterization. One of the simplest parameterization options consists in using a polynomial. This will give us a very simple algorithm, with very few parameters.

Let us consider a K degree polynomial for $g(e)$:

$$g(e) = \sum\nolimits_{k=0}^{K} a_k e^k. \tag{4}$$

Then, its derivative with respect to e is:

$$g'(e) = \sum\nolimits_{k=0}^{K} a_k k e^{k-1} = \mathbf{a}^T\mathbf{e}\,, \tag{5}$$

where $\mathbf{a} = (a_1, a_2, \ldots, a_K)^T$ and $\mathbf{e} = (1, 2e, \ldots, Ke^{K-1})^T$. By using eq (5) into eq (3) we arrive at the following expression for the entropy:

$$H(Z) = H(E) + \mathrm{E}[log(|\mathbf{a}^T\mathbf{e}|)] \tag{6}$$

And the gradient of this expression with respect to the vector of parameters $\mathbf{a}$ (polynomial coefficients) is:

$$\nabla_{\mathbf{a}} H = \mathrm{E}\left[\frac{\mathbf{e}}{\mathbf{a}^T\mathbf{e}}\right], \tag{7}$$

where, by assuming that underlying random processes are wide-sense stationary and ergodic, the expectation can be computed by averaging over time, i.e. by a sample mean estimator.

Finally, we propose an iterative constrained gradient algorithm to estimate the inverse mapping g by repeating the following steps until a convergence criterion or maximum number of iterations is reached:

$$\begin{aligned} \mathbf{a} &\leftarrow \mathbf{a} + \mu\nabla_{\mathbf{a}} H \quad &&\text{; steepest ascend step} \\ \mathbf{a} &\leftarrow \mathbf{a}/_{\sigma_Z} &&\text{; enforce unit} - \text{variance,} \end{aligned} \tag{8}$$

where μ is the stepsize parameter and σ_Z is the standard deviation of the compensated signal $Z = g(E)$.

Polynomial parameterization is very simple, but may have problems when is used for inverting functions whose inverses are not well approximated by a low order polynomial. To avoid working with high order polynomials, in the following section, we propose a nonlinearity parameterization based on a neural network.

2.2.3. Neural network parameterization. Another interesting possibility is to use neural networks, and specifically multi-layer perceptrons (MLP) [16]. As MLP is well known for approximating any continuous and bounded function, it is a good candidate for estimating the inverse function of f, if it exists.

The model of $g(e)$ using a multilayer perceptron with one hidden layer of K units can be written as follows:

$$g(e) = \sum_{k=1}^{K} a_k \sigma(c_k e - b_k), \tag{9}$$

where a_k, b_k and c_k are the weight of the output, the bias and the weight of the input parameters for each unit of the neural network, respectively, and $\sigma(t) = (1 + e^{-t})^{-1}$ is a sigmoid function [16]. Then, its derivative with respect to e is:

$$g\prime(e) = \sum_{k=1}^{K} a_k c_k \sigma\prime(c_k e - b_k) = (\mathbf{a} \circ \mathbf{c})^T \boldsymbol{\theta}, \tag{10}$$

Where $\mathbf{a} = (a_1, a_2, \ldots, a_K)^T$, $\mathbf{b} = (b_1, b_2, \ldots, b_K)^T$, $\mathbf{c} = (c_1, c_2, \ldots, c_K)^T$, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)^T$ with, $\theta_k = \sigma'(c_k e - b_k)$ and '$\circ$' stands for the entry-wise (Hadamard) product of vectors. By using eq (10) in eq (3) we obtain the following expression of the entropy:

$$H(Z) = H(E) + \mathrm{E}[log(|(\mathbf{a} \circ \mathbf{c})^T \boldsymbol{\theta}|)], \tag{11}$$

whose gradients with respect to **a**, **b** and **c** are:

$$\begin{aligned} \nabla_{\mathbf{a}} H &= \mathrm{E}\left[\frac{\mathbf{c} \circ \boldsymbol{\theta}}{(\mathbf{a} \circ \mathbf{c})^T \boldsymbol{\theta}}\right], \\ \nabla_{\mathbf{b}} H &= \mathrm{E}\left[\frac{\mathbf{a} \circ \mathbf{c} \circ \boldsymbol{\varphi}}{(\mathbf{a} \circ \mathbf{c})^T \boldsymbol{\theta}}\right], \\ \nabla_{\mathbf{c}} H &= \mathrm{E}\left[\frac{\mathbf{a} \circ \boldsymbol{\theta} + \mathbf{a} \circ \mathbf{c} \circ \boldsymbol{\varphi}\, e}{(\mathbf{a} \circ \mathbf{c})^T \boldsymbol{\theta}}\right], \end{aligned} \tag{12}$$

where $\boldsymbol{\varphi} = (\varphi_1, \varphi_2, \ldots, \varphi_K)^T$ with $\varphi_k = \sigma''(c_k e - b_k)$. Finally, we propose to use a constrained gradient ascend algorithm as follows:

$$\begin{aligned} &\mathbf{a} \leftarrow \mathbf{a} + \mu_{\mathbf{a}} \nabla_{\mathbf{a}} H && ;\ \text{steepest ascend step for vector } \mathbf{a} \\ &\mathbf{b} \leftarrow \mathbf{b} + \mu_{\mathbf{b}} \nabla_{\mathbf{b}} H && ;\ \text{steepest ascend step for vector } \mathbf{b} \\ &\mathbf{c} \leftarrow \mathbf{c} + \mu_{\mathbf{c}} \nabla_{\mathbf{c}} H && ;\ \text{steepest ascend step for vector } \mathbf{c} \\ &\mathbf{a} \leftarrow \mathbf{a}/_{\sigma_Z} && ;\ \text{enforce unit} - \text{variance}, \end{aligned} \tag{13}$$

where $\mu_{\mathbf{a}}$, $\mu_{\mathbf{b}}$ and $\mu_{\mathbf{c}}$ are the stepsize parameters and σ_Z is the standard deviation of the compensated signal $Z = g(E)$.

3. Experimental Results and Discussion

In order to evaluate the results obtained with the new proposed method, in this section we compute the performances obtained with the Gaussianization method [13], the Uniformization method (similar to [13] but enforcing uniform distribution for Z) and MaxEnt (the new proposed algorithm). The inversion performance for all the methods is calculated as the Signal to

Noise Ratio (SNR), which is defined as follows:

$$SNR = 10\log\left(\frac{\mathrm{E}[z^2]}{\mathrm{E}[(z-x)^2]}\right). \tag{14}$$

It is noted that our gradient search optimization method with both types of parameterizations, i.e. polynomial and neural network, may suffer from stacking at local minima. To alleviate the local minima problem, the initialization for vector **a** (polynomial case) and vectors **a**,**b**,**c** (neural network case) are chosen such that the initial guess of the inversion function is the identity, i.e. $\hat{g}(x) = x$. For example, the initialization in the polynomial parameterization case is as follows: $a_1 = 1$, and $a_k = 0\ \forall\ k \neq 1$.

Taking into account that we can deal with many different scenarios, in the following sections we explore how different conditions and/or values of the parameters can affect the results of the nonlinear compensation. We analyze the effect of the number of samples (T), the mixing matrix (**A**), the number of sources (N) and the nonlinearity type. For the sake of simplicity, MaxEnt only with polynomial parameterization along with Gaussianization and Uniformization are used in **Sections 3.1–3.4**. In all these experiments, the maximum number of iterations for the MaxEnt algorithm is $N_{iter\_max} = 100$. In **Section 3.5**, we present a detailed analysis of the performance obtained by the MaxEnt algorithm with polynomial and neural network parameterizations. In **Section 3.6**, we compare MaxEnt against a *state-of-the-art* methods and, in **Section 3.7**, we experimentally evaluate the sufficient condition of Theorem 1 for different source distributions and nonlinearities.

3.1. Effect of the number of samples (*T*)

In order to evaluate the effect of the number of available samples T, we make experiments by fixing the order of the polynomial to $K = 10$ and by using a random matrix $\mathbf{A}\epsilon\mathbb{R}^{2\times2}$. Sources were generated using zero-mean and unit-variance continuous uniformly distributed random variables, which determines the $[-\sqrt{3}, +\sqrt{3}]$ support. Nonlinearities $f_1(x) = f_2(x) = tanh(3x) + 0.1x$ are used. When we use a mixing matrix **A** the support is increased, therefore there is a strong effect of the nonlinearities. The number of samples ranged from $T = 10$ to $T = 1{,}000$. The experiments were repeated 100 times and the mean and standard deviation of SNRs were computed. In terms of mean values, as we can see in Fig 2a, the MaxEnt algorithm outperforms the Gaussianization (approx. +3dB) and the Uniformization (approx. +6dB) methods. Moreover, MaxEnt algorithm reaches the maximum SNR value (approx. 23dB) quicker than the other two algorithms. It is noted that the variance decreases with large T for all methods. Interestingly, for a very small number of samples $T < 100$, Gaussianization has lower variance than MaxEnt.

In the following experiments we fix the number of samples to $T = 500$.

3.2 Effect of the mixing matrix (A)

Another important parameter is the (unknown) mixing matrix **A**. In order to analyze how the mixing matrix affects the performance of the nonlinear compensation, we parameterize a general mixing matrix **A**, for the case of two sources/observations, as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \tag{15}$$

We did experiments by tuning the parameter ρ from $-0.6 < \rho < +0.6$, for a fixed number of samples $T = 500$, using uniformly distributed continuous random variables as source

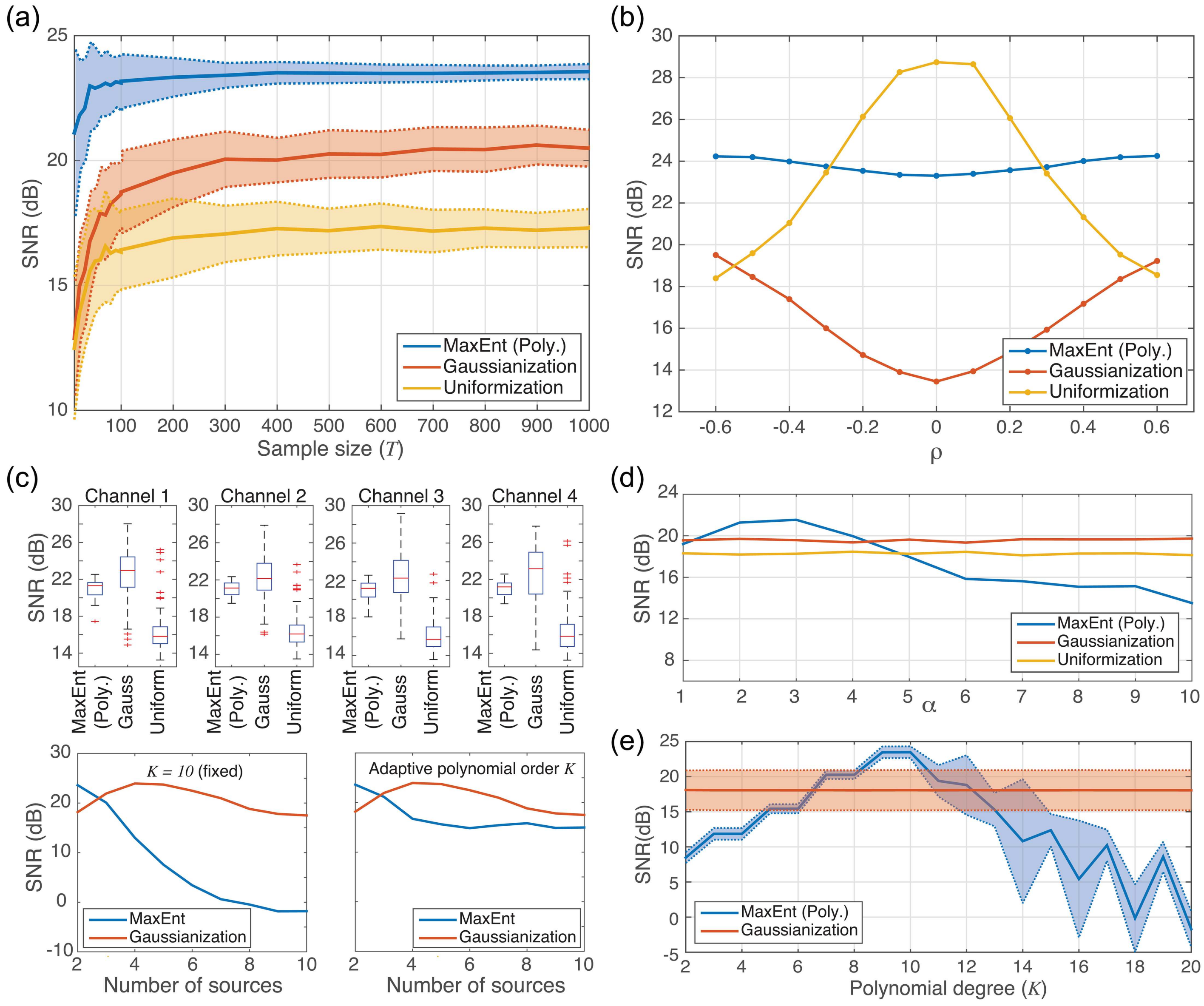

Fig 2. Experimental results with MaxEnt algorithm with polynomial (Poly.) and neural network (N.N.) parameterizations, Gaussianization and Uniformization methods, for different scenarios over a set of 100 experiments: (a) Mean ± std SNR evolution (channel 1) when the number of samples T changes from 10 to 1000 for a fixed random matrix **A**. (b) Mean SNR evolution (channel 1) when the 2 × 2 mixing matrix **A** changes according to parameter ρ as in eq (14). (c) Top: Boxplot of SNR for each channel with a fixed 4 × 4 random mixing matrix. Bottom: SNR as a function of the number of mixing sources for a fixed polynomial order $K = 10$ (left) and for adaptively chosen polynomial order K. (d) Mean SNR versus nonlinearity parameter α as in eq (15). (e) Mean ± std SNR of MaxEnt (Poly.) as a function of the polynomial degree (K) compared against the SNRs obtained with the Gaussianization algorithm.

doi:10.1371/journal.pone.0165288.g002

signals, setting the order of the polynomial to $K = 10$ and fixing the nonlinearities as $f_1(x) = f_2(x) = tanh(3x) + 0.1x$. The obtained averaged SNRs in channel 1, over 100 repeated experiments, are shown in Fig 2b. It is noted that for $\rho = 0$ there is no mixing process, thus the observed signals are directly the distorted (through $f(x)$) original sources. We observe that MaxEnt is stable along the changes on the matrix **A**. On the other hand, it is also interesting to notice the performance of the Gaussianization and Uniformization algorithms: when there is (almost) no mixture ($|\rho|$ close or equal to 0), Gaussianization does not work but

Uniformization does because sources have an exact uniform distribution. However, it is important to note that in practice sources are not uniform, hence for practical situations Uniformization would not give such perfect solutions.

3.3 Results with $N \geq 4$ sources

Until now and for the sake of simplicity, we have presented results of the inversion of nonlinearities for the case of mixtures with only two sources. If we have more mixing sources, we tend to a better situation according to the Central Limit Theorem, therefore we should expect good results for Gaussianization algorithm. First, we performed experiments mixing $N = 4$ random uniformly distributed continuous sources and using 100 different (random) mixing matrices $\mathbf{A}$ (with ones in the main diagonal). We fixed the order of the polynomial to $K = 10$ and nonlinearities as $f(x) = tanh(3x) + 0.1x$. Fig 2c **(top)** presents the obtained SNRs (dB) for all 100 experiments in each channel as a box plot. We can see that Gaussianization algorithm obtains best-averaged results (about 23 dB in channel 1) but with a very high variance (about 2.5 dB), while MaxEnt provides an average performance a little bit lower ($<$ 1 dB less), with a very small variance (about 1 dB). So, the MaxEnt algorithm is the best compromise. It is highlighted that, by using the Gaussianization and Uniformization methods, the results have a considerably larger variance compared to the MaxEnt algorithm. In other words, MaxEnt is more robust because the algorithm maximizes entropy independently of the distribution of the mixture. The Uniformization method gives the worst results because mixing more sources makes the mixture to be farther away from the uniform distribution.

Finally, we analyzed the performance of MaxEnt and Gaussianization as a function of the number of mixing sources, with the nonlinearity $f(x) = tanh(3x) + 0.1x$. and using 100 different (random) mixing matrices $\mathbf{A}$ (with ones in the main diagonal). Fig 2c **(bottom-left)** shows the obtained SNRs (dB) for the case of using a fixed degree of the polynomial to $K = 10$, whereas in Fig 2c **(bottom-right)** the adaptive strategy for setting the polynomial degree K was used (see section 3.4). It is noted that there is a trade-off between Gaussianization and MaxEnt. For more input sources, Gaussianization is better since polynomial limit the accuracy of Maxent and, for less and non-Gaussian input sources, Gaussianization is less precise.

3.4 Effect of the nonlinearities

The type of nonlinearity plays also an important role. Using polynomials, it is expected that our method will be successful only when the inverse of the monotonic nonlinear function is well approximated by a finite degree polynomial. Here, we explore how the polynomial parameterization deals with a family of functions of the form

$$f(x) = tanh(\alpha \cdot x) + 0.1x\,, \quad \alpha = 1,\ 2,\ \ldots,\ 10. \tag{16}$$

For $\alpha = 1$ the nonlinear function is almost linear, while for $\alpha \geq 5$ the nonlinear function is highly saturating the input signal (see Fig 3). Therefore, the inverse will be a polynomial of low degree in the first case and a polynomial of higher degree in the second case. To deal with this optimal selection of the polynomial, we have computed the objective function $\mathrm{E}[log(|\mathbf{a}^T\mathbf{e}|)]$ (eq (6)) for a wide range, e.g. degree 2 to degree 15, and selected the best degree value K for the polynomial i.e. the one that provides largest entropy. In Fig 2c (**bottom**), we compare the results of using a fixed polynomial degree ($K = 10$, left) against using the dynamic selection of polynomial degree (right) as a function of the number of mixing sources.

Fig 2d shows the results for MaxEnt, Gaussianization and Uniformization for all types of nonlinearities of eq (16). It is noted that MaxEnt outperforms the rest of algorithms for $2 \leq \alpha \leq 4$. When $\alpha = 1$ the performance of MaxEnt (mean SNR = 19.21dB, std = 0.26) is comparable

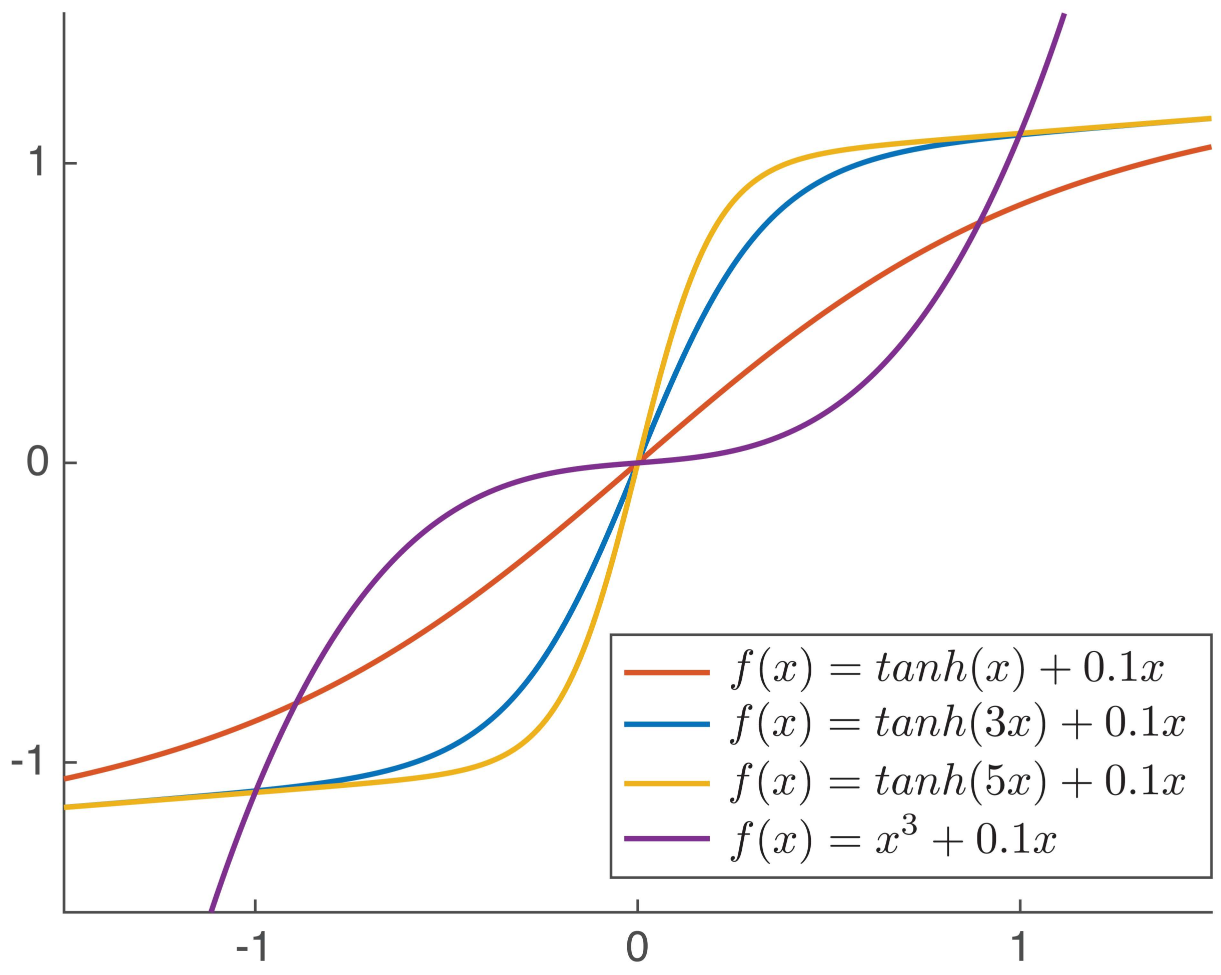

Fig 3. Family of nonlinearities used in the experiments.

doi:10.1371/journal.pone.0165288.g003

with Gaussianization (mean SNR = 19.56, std = 0.97). However, when $\alpha > 4$ it the performance of MaxEnt drops because the polynomial parameterization is not able to adapt to the high-saturating shape of the distortion. It is noted also that the degree of the polynomial selected by the algorithm changes according to the complexity of the function to be inverted: for smooth functions ($\alpha = 1$) the selected degree is small ($K = 4$), while for a harder distortion ($\alpha = 5$) the selected degree is higher ($K = 12$). According to these results, in the following sections, we fix the parameter $\alpha = 3$ and the degree of the polynomial to $K = 10$.

3.5. Effect of sources distributions and nonlinearities on the MaxEnt performance with polynomial and neural network parameterization

In this set of experiments we explore the performance obtained by MaxEnt algorithm with polynomial and neural network parameterizations for two different types of nonlinearities: $f(x) = tanh(3x) + 0.1x$ and $f(x) = x^3 + 0.1x$ (see Fig 2). More specifically, we compare our approach against the Gaussianization and Uniformization methods using two types of sources: uniformly distributed continuous random noise and (3-level) discrete random noise (−0.4, 0, +0.4). Based on the analysis of the previous set of experiments, we fix the parameters according to Table 1.

Table 1. Parameter selection for experiments with polynomial parameterization (1 case) and with neural network parameterization (2 cases).

	Polynomial	Neural network (I)	Neural network (II)
Mixing matrix	$\boldsymbol{A} = \begin{bmatrix} 1 & 0.6 \\ 0.7 & 1 \end{bmatrix}$	$\boldsymbol{A} = \begin{bmatrix} 1 & 0.6 \\ 0.7 & 1 \end{bmatrix}$	$\boldsymbol{A} = \begin{bmatrix} 1 & 0.6 \\ 0.7 & 1 \end{bmatrix}$
Nonlinear functions	$f(x) = tanh(3x) + 0.1x$	$f(x) = tanh(3x) + 0.1x$	$f(x) = x^3 + 0.1x$
Sample size	$T = 500$	$T = 500$	$T = 500$
Model order	$K = 10$	$K = 7$	$K = 7$
Maximum number of iterations	$N_{iter\_max} = 100$	$N_{iter\_max} = 50{,}000$	$N_{iter\_max} = 50{,}000$
Adaptation step	$\mu = 0.1$	$\mu = 10$	$\mu = 0.001$
Sources type	Discrete or continuous (uniformly distributed)	Discrete or continuous (uniformly distributed)	Discrete or continuous (uniformly distributed)
Number of experiments	$N_{exp} = 100$	$N_{exp} = 20$	$N_{exp} = 20$

doi:10.1371/journal.pone.0165288.t001

We have repeated the experiments 100 times for the polynomial parameterization and 20 times for the neural network parameterization. Fig 4 shows statistics of SNR improvement in dB obtained with all the methods in all the situations. In this figure we observe that MaxEnt with polynomial parameterization gives always the best SNR (dB) (Fig 4a) having smaller variance in the case of continuous sources compared to the rest of the methods. On the other hand, neural network parameterization gives similar results as Gaussianization and Uniformization for the case of continuous sources but it outperforms those methods for the case of discrete sources (Fig 4a and 4b), especially for when $f(x) = x^3 + 0.1x$. It is noted that discrete sources do not work as well as continuous sources under the Gaussianization because a summation of two 3-state random variables results in a 7- to 9-state random variable, and this will hardly approximate a Gaussian random variable when only two variables are summed. It is a mere consequence of the asymptotic character of the central limit theorem that is not met in this setting; hence maximal entropy methods should succeed better.

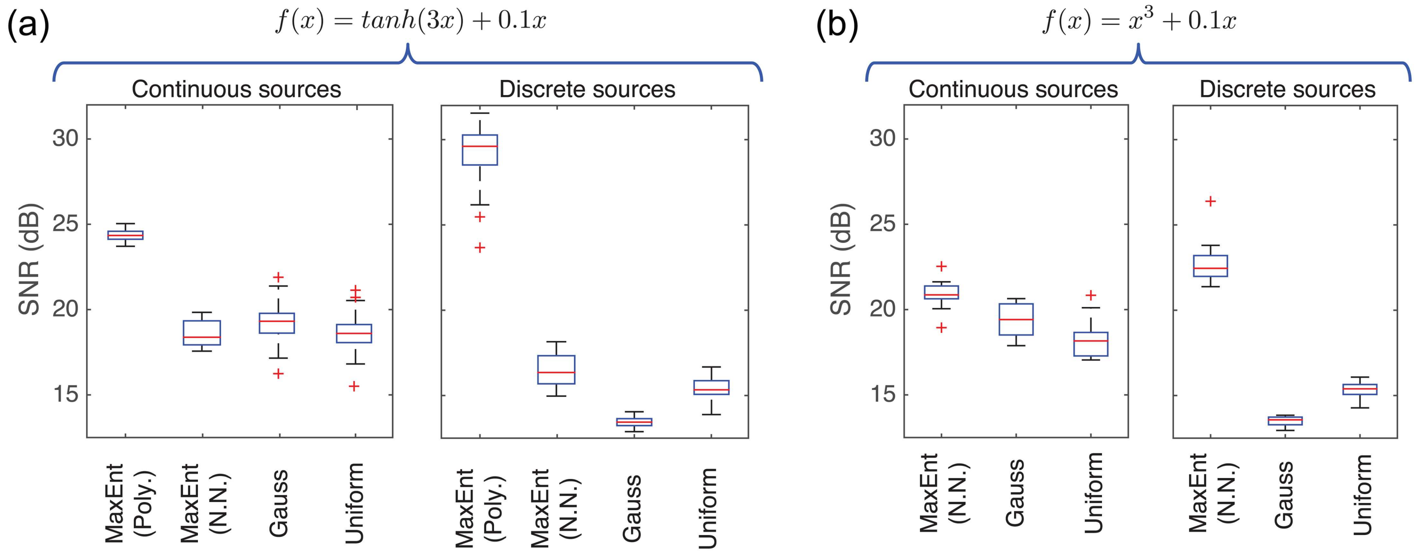

Fig 4. SNR (dB) boxplots obtained in channel 1 for MaxEnt algorithm with polynomial (Poly.) and neural network (N.N.) parameterizations, Gaussianization and Uniformization methods, for continuous (uniformly distributed) and (3-level) discrete sources (–0.4, 0, +0.4). (a) Results for the nonlinearity $f(x) = tanh(3x) + 0.1x$. (b) Results for the nonlinearity $f(x) = x^3 + 0.1x$ (polynomial parameterization is not included here because MaxEnt was not able to well approximate the nonlinearity showing convergence problems).

doi:10.1371/journal.pone.0165288.g004

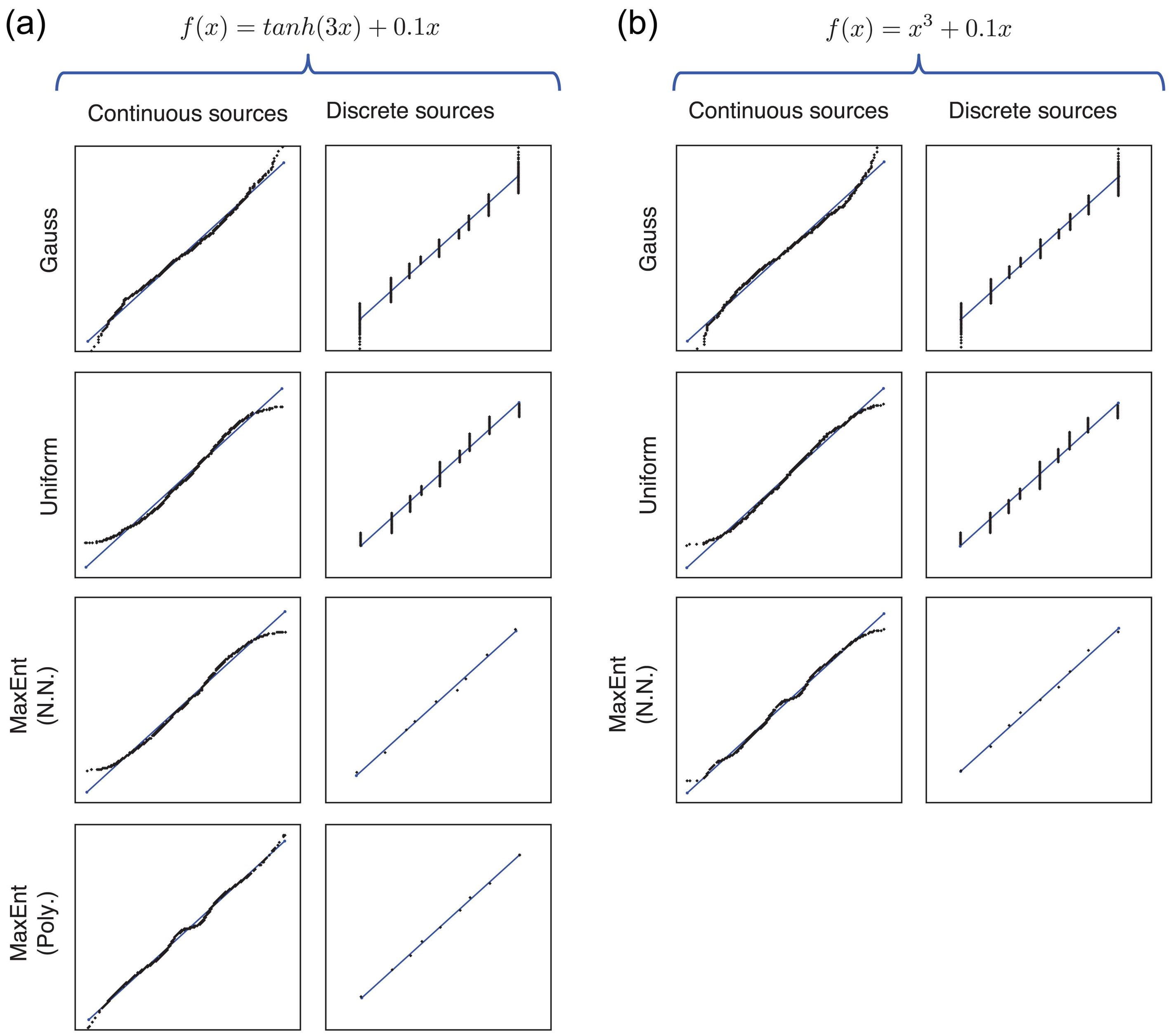

Fig 5. Examples of obtained nonlinear compensation for each of the tested algorithms for continuous (uniformly distributed) and (3-level) discrete sources (–0.4, 0, +0.4). (a) Results for the nonlinearity $f(x) = tanh(3x) + 0.1x$. (b) Results for the nonlinearity $f(x) = x^3 + 0.1x$.

doi:10.1371/journal.pone.0165288.g005

It is important to note that the polynomial parameterization cannot approximate well the inverse of the nonlinear function $f(x) = x^3 + 0.1x$, so the MaxEnt algorithm didn't converge resulting in erroneous solutions. For this reason, MaxEnt (Poly) was not included in Fig 4b. On the other hand, neural network parameterization works very well for this nonlinearity outperforming the Uniformization and Gaussianization methods.

Examples of the nonlinear compensations obtained by all the methods in these experiments are shown in Fig 5 in the form of $x(t)$ versus $z(t)$ scatter-plot. In order to visually compare the performance, the identity function is superimposed to each curve. It is interesting to note that

Table 2. SNR (dB) as mean±std obtained in 100 experiments. Nonlinear part and global system inversion results are presented, for MaxEnt+JADE and MIM algorithms.

	Non-linear part inversion	Non-linear part inversion	Global system inversion	Global system inversion
	New method	**MIM method**	**New method**	**MIM method**
Channel 1	24.30±0.32	21.48±1.57	17.22±0.39	16.85±1.99
Channel 2	24.00±0.37	20.74±1.37	16.29±0.61	15.91±2.40

doi:10.1371/journal.pone.0165288.t002

Gaussianization, Uniformization and MaxEnt with neural networks provide less accurate compensations at the edges, while MaxEnt with polynomial parameterization does in the central region.

3.6 Comparison of MaxEnt against a state-of-the-art global optimization based algorithm

In order to demonstrate the validity of our algorithm, we present here some simulation results comparing our method combined with the classical JADE algorithm [17], the MaxEnt+JADE algorithm (JADE Matlab code available in the author's webpage: http://perso.telecom-paristech.fr/~cardoso/Algo/Jade/jadeR.m), against Mutual Information Minimization (MIM) algorithm proposed in [7] (MIM Matlab code provided by Prof. Christian Jutten, GIPSA Lab.), which implements a global optimization, i.e. by inverting at the same time the nonlinear and linear parts. We used the same parameters as in section 3.5 (see Table 1, Polynomial case), for uniformly distributed continuous sources.

Numerical SNR (dB) results as (mean±std) are shown in Table 2. We can observe that both strategies are very similar in terms of the obtained global mean SNR, but our new method is more robust (smaller variance). Also, it is interesting to note that our algorithm is much better than MIM regarding the inversion of non-linear part, so we suspect that the global performance could be further improved by using other algorithm for the linear part, for which there are a bunch of alternatives in the literature.

Regarding the associated computational cost, we obtained an improvement of one order of magnitude. More specifically, in this example, our method requires 0.28 seconds in average for inverting the global system, while MIM method spent 2.86 seconds in average. MIM computational time increases exponentially with T while our algorithm has a linear dependency on T, so the difference will be higher with longer signals.

In Fig 6 we present an example of nonlinear compensation using our MaxEnt method (blue line) and MIM (red line), for channel 1 (left) and channel 2 (right).

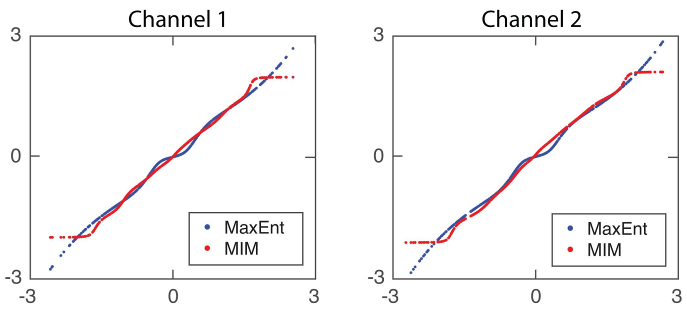

Fig 6. Nonlinear compensation of nonlinearities for MaxEnt algorithm (blue line) and MIM algorithm (red line). Channel 1 is depicted in the left part, channel 2 in the right part.

doi:10.1371/journal.pone.0165288.g006

Table 3. Experimental evaluation of the sufficient condition in Theorem 1. T = 10,000 samples where used and experiments where repeated N = 500 times. The significance p value (H0: $E^2[f'(X) > E[f^2(X)])$ in all the cases where $p \ll 0.0001$. The fourth and sixth order moments were computed also experimentally. Cases for which the sufficient condition does not hold true are displayed in bold type (< 0).

	Gauss	$x_1 + x_2$, x_p **unif.**	$x_{1'}^2 + x_{2'}^2$, x_p **unif.**	$x_{1'}^3 + x_{2'}^3$, x_p **unif.**
μ_4	3.0	2.4	2.5	2.8
μ_6	14.9	7.5	10.4	15.3
$f(x)$		$E[f^2(X)]-E^2[f'(X)]$		
$tanh(x) + 0.1x$	> 0	> 0	> 0	> 0
$tanh(3x) + 0.1x$	> 0	> 0	> 0	> 0
$tanh(5x) + 0.1x$	> 0	> 0	> 0	> 0
$x^3 + 0.1x$	> 0	**< 0**	> 0	> 0
$x^3 + 0.3x$	> 0	**< 0**	> 0	> 0
$x^3 + 0.5x$	> 0	**< 0**	> 0	> 0

doi:10.1371/journal.pone.0165288.t003

Finally, it is important to note also that our nonlinear inversion strategy allow us to use any linear BSS algorithm. We have only tried with JADE but any other algorithm can be used, and this could potentially increase the final performance.

3.7 Experimental evaluation of the sufficient condition in Theorem 1

In Table 3, we evaluated experimentally the condition of eq (1) by averaging $f^2(x)$ and $f'(x)$ over T = 10,000 samples, using several probability distributions $p_X(x)$ and various nonlinearities. As expected, the condition holds true for the case of a Gaussian distribution (second column). Also, as expected the condition does not hold true for the case of a variable X being the sum of two uniformly distributed variables (third column) and $f(x) = x^3 + \beta x$, because in this case $\mu_4 < 3$ and $\mu_9 < 9$ (see Corollary 1. Case II). However, our experimental results in section 3.5, shows that even when the condition does not hold true in this case, the MaxEnt method successfully invert the nonlinear transform.

Besides, we generated also samples for a variable X being the sum of squared and cubic uniformly distributed variables in order to have a wider range of fourth and sixth order moments (fourth and fifth columns). In all these cases the sufficient condition held true, which gives us a guarantee that the MaxEnt works in these cases.

Conclusions

In this paper, we proposed a generalization of the Gaussianization/Uniformization method for blind estimation of a nonlinear map. The method is based on the entropy maximization of the nonlinear outputs in order to approximate the unknown nonlinear function. We provide a sufficient condition on the probability distribution and the nonlinear distortion that gives a guarantee for the MaxEnt method to succeed inverting the nonlinearity and analyze it in detail for a particular case of nonlinear distortion.

In order to maximize the entropy of the observations, we introduced two different parameterization strategies based on polynomial and neural network parameterizations, respectively, and we developed constrained gradient steepest ascend MaxEnt algorithms.

By an extensive experimental set, we explored several possible scenarios, analysing the effect of the type of the sources (uniform or discrete random noise), the length of the sources (T), the mixing matrix $\mathbf{A}$, the nonlinearities and the number of sources (N). We also performed experiments comparing both types of proposed parameterizations, i.e. polynomial and neural network, in order to show pros and cons for each one.

Polynomial parameterization is a very good option when the function to be compensated can be inverted with a polynomial of low order (less than $K = 12$). In this case, the method is extremely fast and converges in a very few number of iterations. Moreover, the order of the inverse function (polynomial) can be adjusted automatically by evaluating the objective function $\mathrm{E}[\log(|\mathbf{a}^{\mathrm{T}}\mathbf{e}|)]$, as detailed in eq (6). This allows overcoming the problem of tuning parameters, making the method easier to use. On the other hand, neural network parameterization obtains similar results as Gaussianization or Uniformization for continuous uniform random noise, and overcomes those methods for discrete random noise sources. The main advantage of this parameterization is that it can deal with nonlinear functions that are not invertible with a polynomial, obtaining in this case better results than Gaussianization or Uniformization (see Fig 4b). The main drawback of neural network parameterization is that the inversion of nonlinearities is much slower than the polynomial case, about 1000 times slower.

If the number of sources is high, or one source is known to be Gaussian in the case of 2 sources, Gaussianization method works equal or better than MaxEnt because the linear mixture is Gaussian. But in practical situations we do not have access to prior information about the sources, therefore MaxEnt with polynomial parameterization is a good option.

As demonstrated by the experiments, maximizing entropy allow us to decouple estimations of the nonlinear part from the linear one. Therefore, in the PNL source separation scenario, we can apply this strategy to linearize the system and revert the problem to a (linear) blind source separation scenario. As the nonlinearities are inverted independently in each channel, this part can be solved in parallel. Then, any source separation algorithm for linear mixture can be used in order to recover the original observations. This procedure can also be applied for solving Wiener systems that, as it has been explained above, are equivalent to PNL mixtures and therefore can be processed with similar techniques.

Further work can be done, especially concerning the choice of the parametric form for the nonlinear functions. Polynomials are interesting for their simplicity and the linearity with respect to the parameters. For many monotonic functions only a few parameters are required, but the number of parameters can increase dramatically for functions with very large slopes. Then, splines or radial basis functions (RBF) could be a good alternative to consider.

Appendix: Proofs

Proof of Theorem 1

We need to prove that, under the condition of eq (1), *the following equation holds true:*

$$H(\bar{E}) < H(X) \tag{17}$$

where $\bar{E}$ is the normalized version of variable E, i.e. $\bar{E} = E/\sigma_E$, with $\sigma_E = \sqrt{\mathrm{E}[E^2]}$ is the standard deviation of the output variable E. By using a property of the entropy we obtain

$$H(\bar{E}) = H(X) + \mathrm{E}[\log(f'(X))] - \log(\sigma_E). \tag{18}$$

Thus, by proving that $E[log(f'(X))] < log(\sigma_E)$ will suffice to prove this theorem.

The condition of eq (1) *implies that*

$$\begin{gathered} \mathrm{E}[f'(X)] < \sigma_E \\ \log(\mathrm{E}[f'(X)]) < \log(\sigma_E), \end{gathered} \tag{19}$$

and, by applying the Jensen inequality to the last line in eq (19), we arrive at:

$$\mathrm{E}[\log(f'(X))] < \log(\sigma_E) \tag{20}$$

which implies that $\mathrm{E}[\log(f'(X))] < \log(\sigma_E)$ *and completes the proof.*

Proof of Corollary 1

Here, we study the conditions under which eq (1) holds true, for the particular case of having the nonlinear transform $f(x) = x^3 + \beta x$. *In this case, it is straightforward to evaluate the left and right hands of eq (1) as follows:*

$$E^2[f'(X)] = \mathrm{E}^2[3X^2 + \beta] = (3 + \beta)^2 \tag{21}$$

and

$$\mathrm{E}[f^2(X)] = \mathrm{E}[X^6 + 2\beta X^4 + \beta^2 X^2] = \mu_6 + 2\beta\mu_4 + \beta^2. \tag{22}$$

By putting together the above two equations into eq (1), we obtain the following general condition:

$$2\beta(\mu_4 - 3) + (\mu_6 - 9) > 0. \tag{23}$$

This equation states the conditions on the moments μ_4, μ_6 *and the parameter* β *that make the sufficient condition of eq (1) holds true.*

Acknowledgments

This work has been partly funded by the University of Vic—Central University of Catalonia under the grant R0947 and grants from ANPCyT—PICT 2012–1519 and CONICET—PIP 114-201101-00021, Argentina. The authors would like to acknowledge Prof. Christian Jutten (GIPSA Lab, Grenoble, France) for his discussions and guidance across this research work. The authors thank reviewers for providing deep and useful comments on the first version of the manuscript.

Author Contributions

Conceptualization: JSC CFC.

Formal analysis: JSC KLP.

Investigation: JSC CFC KLP.

Methodology: JSC CFC.

Project administration: JSC CFC.

Software: JSC KLP.

Supervision: JSC CFC.

Visualization: JSC CFC.

Writing – original draft: JSC CFC KLP.

Writing – review & editing: JSC CFC.

References

1. Chi CY, Feng CC, Chii-Horng C, Chen CY. Blind Equalization and System Identification: Batch Processing Algorithms, Performance and Applications. Springer London; 2006.

2. Bai E-W. Identification of Linear Systems with Hard Input Nonlinearities of Known Structure. IFAC Proceedings Volumes. 2002; 35: 97–102.

3. Nikias CL, Petropulu AP. Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework. 1993. PTR Prentice Hall, Englewood Cliffs, NJ.

4. Astola JT, Egiazarian KO, Khlopov GI, Khomenko SI, Kurbatov IV, Morozov VY, et al. Application of Bispectrum Estimation for Time-Frequency Analysis of Ground Surveillance Doppler Radar Echo Signals. IEEE Trans Instrum Meas. 2008; 57: 1949–1957.

5. Bai Er-Wei, Li Qingyu, Zhang Zhiyong. Blind source separation/channel equalization of nonlinear channels with binary inputs. IEEE Trans Signal Process. 53: 2315–2323.

6. Bai EW, Cerone V, Regruto D. Separable inputs for the identification of block-oriented nonlinear systems. 2007 American Control Conference. 2007. pp. 1548–1553.

7. Taleb A, Jutten C. Source separation in post-nonlinear mixtures. IEEE Trans Signal Process. 1999; 47: 2807–2820.

8. Bai E-W, Cai Z, Dudley-Javorosk S, Shields RK. Identification of a Modified Wiener-Hammerstein System and Its Application in Electrically Stimulated Paralyzed Skeletal Muscle Modeling. Automatica. 2009; 45: 736–743. doi: 10.1016/j.automatica.2008.09.023 PMID: 23467426

9. Sano M, Sun L. Identification of Hammerstein-Wiener system with application to compensation for nonlinear distortion. SICE 2002 Proceedings of the 41st SICE Annual Conference. 2002. pp. 1521–1526 vol.3.

10. Hunter IW, Korenberg MJ. The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. Biol Cybern. 1986; 55: 135–144. PMID: 3801534

11. Taleb A, Sole J, Jutten C. Quasi-nonparametric blind inversion of Wiener systems. IEEE Trans Signal Process. 2001; 49: 917–924.

12. Sole J, Jutten C, Taleb A. Parametric approach to blind deconvolution of nonlinear channels. Neurocomputing. 2002; 48: 339–355.

13. Solé-Casals J, Jutten C, Pham DT. Fast approximation of nonlinearities for improving inversion algorithms of PNL mixtures and Wiener systems. Signal Processing. 2005/9; 85: 1780–1786.

14. Huang D, Huang X, Leung H. Nonlinear Compensation of High Power Amplifier Distortion for Communication Using a Histogram-Based Method. IEEE Trans Signal Process. 2006; 54: 4343–4351.

15. Ziehe A, Kawanabe M, Harmeling S, Müller K-R, Blind Separation of Post-Nolinear Mixtures using Gaussianizing Transformations and Temporal Decorrelation, Proceedings of the 4th International Symposium on Independent Component Analysis (ICA2003), Nara (Japan), April 2003, pp. 269–274.

16. Haykin SS. Neural Networks and Learning Machines. Prentice Hall; 2009.

17. Cardoso JF. High-order contrasts for independent component analysis. Neural Comput. 1999; 11: 157–192. PMID: 9950728