International Journal of Future Generation Communication and Networking
Vol. 2, No. 3, September, 2009

Improvement to Geometric Linearization of gpICA Algorithm by Compensation and Multiple Points

Erika Torres[1], Paula Ulloa[2], Andrés Gaona[3], and Miguel Torres[4]

[123]*Faculty of Engineering, Distrital University FJC, Colombia*
[4]*Faculty of Engineering, Javeriana University, Colombia*
[1]*akane999@gmail.com,* [2]*paula.ulloa@gmail.com,* [3]*angaona@gmail.com,*
[4]*metorres@javeriana.edu.coAuthor*

Abstract

This paper presents two modifications of the gpICA (geometric post non-linear independent component analysis) algorithm. gpICA algorithm is a novel method to solve the PNL (Post non-linear) scheme. We propose these modifications to improve the mean squared error, the correlation of the recovered signals and algorithm reliability . The first improvement, called compensation, takes advantage of the implicit information given by the point to be linearized. On the other hand, while the original gpICA algorithm uses two sets of two points to make an update, our second modification uses two sets of four points. We present experimental results which validates the effectiveness of each modification. The PNL applications can be seen in sensor array processing, digital satellite, microwave communications, biological systems and nonlinear blind source separation tasks. gpICA can recover the original sources of a nonlinear mixture, unlike some of the other nonlinear BSS algorithms, it does not require any assumption on the distribution of the input signals.

Keywords: *Blind Source Separation (BSS) , Independent Component Analysis (ICA), Post non-linear mixtures, Speech Signal Processing.*

1. Introduction

Blind Source Separation (BSS) is the general problem of determining original sources when only their mixtures are available for observation [1]. The BSS problem has been resolved from different approaches, one of them is the Independent Component Analysis (ICA) [2]. BSS resolved using ICA consist in the recovery of a set of independent signals from a set of observations that are unknown linear mixtures of the independent source signals. When assuming statistical independence between the sources, ICA has shown to have many applications in real world problems, such as in speech recognition systems, telecommunications and medical signal processing.

Many ICA algorithm variations have been proposed to solve the linear case with different approaches (JADE [3], Fast ICA [4], Pearson ICA [5], etc.). However, in many situations, the basic linear model can not describe the real system adequately. For example all microphones can capture only until certain level of power, this phenomenon is called saturation, most sensors have this nonlinearity, such disadvantage can be overcome with a suitable nonlinear model. Nonlinear BSS is a more realistic model, but is more difficult to solve.

In orden to solve a more realistic model, Jutten et. al. proposed an important subclass of nonlinear BSS models called Post Nonlinear (PNL) scheme [6]. The PNL scheme constrains

the transformation to one dimensional invertible nonlinear functions. The main idea behind the PNL scheme consists in finding the inverse of such nonlinear functions, in order to reduce the problem to a linear case. Hence, this linear case can be solved using any ICA algorithm.

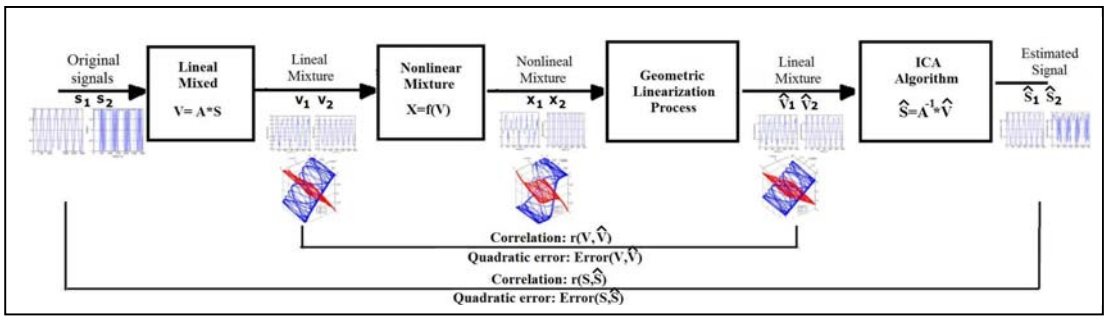

Figure 1. Lineal, nonlineal mixture and gpICA algorithm: geometric linearization and blind source separation (use ICA). A is a 2x2 Matrix and f() is a nonlineal function

The PNL scheme is not a trivial problem, in order to solve it, Nguyen et. al. proposed the gpICA (geometric post nonlinear independent component analysis) algorithm in [7]. The gpICA algorithm was inspired from the next simple observation: in a multidimensional space, the graph of a nonlinear mixture is a nonlinear surface. On the other hand, when the mixture is a linear one, its graph is a plane. Based on this observation, the gpICA algorithm works in two stages. The first stage transforms a nonlinear surface (the geometric representation of the nonlinear mixture) in to a plane (The geometric representation of a linear mixture). The geometric linearization process reduces the nonlinear BSS problem in to a linear one. The second stage applies a linear BSS algorithm to extract the original sources.

The paper is organized as follows: Principles of gpICA algorithm are explained in Section 2. In Section 3 a scheme in 2D for gpICA Algorithm is presented, with this we developed two improvements to the original geometric linearization algorithm (First stage of gpICA): Compensation and Linearization with multiple points, their details are shown in Sections 4 and 5. We provide some computer simulation results in Section 6 in order to validate each modification. Finally, we discuss the issues related to the proposed improvements in Section 7.

2. gpICA: Geometric Post-nonlinear ICA

The gpICA algorithm was proposed by Nguyen et. al. in [7], and was based on the PNL model with a geometric linearization process, followed by an ICA algorithm. The geometric approach has some advantages over the other PNL algorithms: (i) the first stage and second stage are independent from each other, therefore, any ICA algorithm can be applied.(ii) unlike some of the other algorithms, it does not require any assumption on the distribution of the input signals [8].

In a 3D space, with two sources s1 and s2 where values are in axis x-y, respectively. The values of vi (linear mixture) and xi (nonlinear mixture) are in z-axis, forming a plane and a nonlinear surface, respectively. In order to transform a plane from a given nonlinear surface, gpICA uses properties of the straight line in 3D space. This is shown in Figure 1.

The notation " companion points " is given to the points that belong to the 3D plane, which

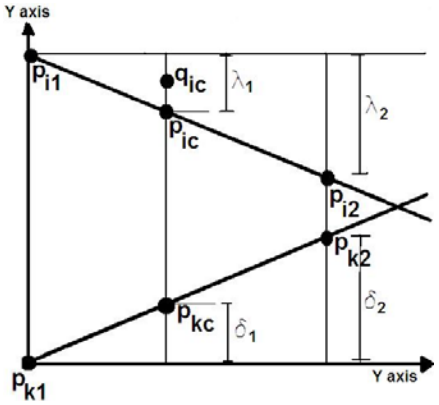

Figure 2. Scheme in 2D for gpICA Algorithm. 2D cut of a 3D lineal mixture planes.

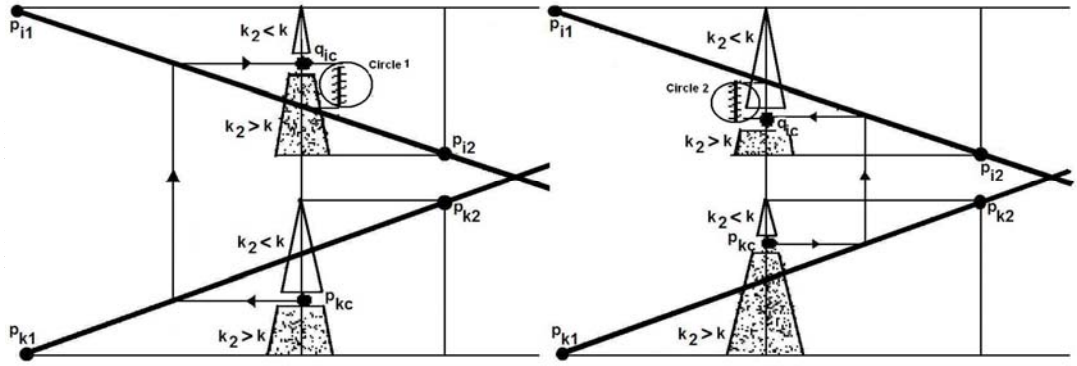

Figure 3. Improvement 1: Linearization with Compensation. Left: Case 1, Right: Case 2. Inside the circle 1 and 2 shown the parts where original algorithm fail

have equal position in x and y coordinate axis (unknown values). To recognize such points a time index to identify a companion point is used, that is, point $p_1 = (v_1(t_1),v_2(t_1),x_1(t_1))$ has a companion point $p_2 = (v_1(t_2),v_2(t_2),x_1(t_2))$.

The algorithm uses a "fake plane" S_k as reference plane and use it to transform the other surfaces S_x. The "fake plane" changes alternatively from one surface to another during the transformation process. The transformation scheme for two sources can be explained by the following steps:

1) Sort the x signals.
2) Filter the sorted x signals using equation (1). cont = 0.
3) Restore the original order of the smoothed x signals.
4) Select fake plane named S_k. Other surface has been called S_i .
5) Pick up two random $p_{k1}=(v_1(t_1),v_2(t_1),x_k(t_1))$ and $p_{k2}=(v_1(t_2),v_2(t_2),x_k(t_2))$ from the surface S_k. Locate their respective " companion points " p_{i1} and p_{i2} in the surface S_i.
6) Select an arbitrary point $p_{kc}=(v_1(t_c),v_2(tc),x_k(t_c))$ between p_{k1} and p_{k2}, and its companion point p_{ic} in S_i . Find the value $z_i(t_c)$ of point $q_{ic}=(v_1(t_c),v_2(t_c), z_i(t_c))$ using (2).
7) Compute $x_{newi}(t_c)$ using equation (3).
8) if cont $\leq N_{k,}$ then cont=cont +1 and go to step 4. Else: cont=0; go to next step.

9) Compute the error using equation (4).
10) if e > x go to numeral 2, Else v = x. finish.

Table 1. Improvement 1: Geometric linearization with Compensation cases

Case 1: q_{ic} over $\overline{p_{i1}p_{i2}}$		Case 2: q_{ic} under $\overline{p_{i1}p_{i2}}$	
$p_{ic} \in$	p_{ic} Position	$p_{ic} \in$	p_{ic} Position
$p_{ic} \in k_2 < k$	correct	$p_{ic} \in k_2 < k$	correct
$p_{ic} \in k_2 > k$	wrong (over $\overline{p_{i1}p_{i2}}$) Circle 1 otherwise good	$p_{ic} \in k_2 > k$	wrong (under $\overline{p_{i1}p_{i2}}$) Circle 2 otherwise good

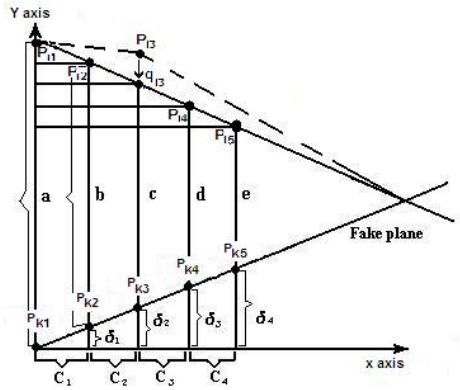

Figure 4. Improvement 2: Geometric linearization with Multiple Points. PNL mixture vs. sources

$$\tilde{z}i(t) = \frac{1}{L}\sum j = -(L-1)/2(L-1)/2zis(t+j) \tag{1}$$

$$zi(tc) = \frac{xk(tc) - xk(t1)}{xk(t2) - xk(t1)}(xi(t2) - xi(t1)) + xi(t1) \tag{2}$$

$$xinew(tc) = \mu zi(tc) + (1-\mu)xiold(tc) \tag{3}$$

$$\varepsilon = \frac{1}{nNk}\sum i = 1n\sum j = 1Nk(xinew(j) - xiold(j))2 \tag{4}$$

Where $\mu \in [0,1]$, ξ is a threshold to stop the linearization and N_k is the number of points to be updated in each iteration. For more sources the same transformation scheme is applied, but executes the actualization (equation (2) and (3)) in every plane different to the fake plane.

3. Scheme in 2D for gpICA Algorithm

The linearization algorithm is easier to understand in a transversal cut over line $\overline{p_{k1}p_{k2}}$ of the 3D plane, obtaining a 2D plane shown in Figure 2, (with the values of x_i in Y axis, (being the only known values). Points pk_1, pk_2, . . ., are transformed in a 2D plane.

We have $\lambda_2=k\delta_2$, where we can compute $k=\lambda_2/\delta_2$. Then we can obtain $\lambda_1=k\delta_1$, and $z_c=x_i(t_1)-\lambda_1$, this is the same equation (2), with $\delta_1=x_k(t_c)-x_k(t_1)$, $\delta_2=x_k(t_2)-x_k(t_1)$, $\lambda_1=x_i(t_1)-x_i(t_c)$ and $\lambda_2= x_i(t_1)-x_i(t_2)$.

4. Improvement 1: Linearization with compensation

The geometric linearization algorithm presents problems when q_{ic} moves away from line $\overline{p_{i1}p_{i2}}$, an explanation of this is shown in Figure 3 and Table 1.

The variable k_2 is $k_2 = \lambda_1/\delta_1$ (see Figure 3), this is the same relationship between k and points $p_{k2}-p_{i2}$, but in points $p_{kc}- p_{ic}$. There are three zones in line $\overline{p_{kc}p_{ic}}$: i) $k_2<k$, ii) $k_2>k$, and iii) $k_2=k$ when we have a line (perfect linearization); this information will be used to compensate the problem with the wrong position for q_{ic}, show in circles in Figure 3.

Figure 3 shows two cases for the point p_{kc}, in the first case (p_{kc} under $\overline{p_{k1}p_{k2}}$). we have that if the original point p_{ic} belongs to the zone $k_2<k$, those cases are shown in detail in the Table 1. Wrong zones are depicted in the Figure 3. A Wrong position occurs when gpICA does not correct the position, on the contrary, damaging information.

The wrong position for q_{ic} is compensated by adding or subtracting the difference $(z_i(t_c)-x_i(t_c))$. The compensation in a zone around p_{ic} is greater than the compensation applied to a point outside this zone, then we multiply the difference by k_2/k. The new $z_i(t_c)$ compensated $z_i(t_c)$ named $z_i(tc)^{cp}$ shown in equation (5) is obtained of the $z_i(t_c)$ calculated in (2).

$$k_2 < k : z_i(t_c)^{cp} = z_i(t_c) + (z_i(t_c) - x_i(t_c))\frac{k_2}{k}$$
$$k_2 > k : z_i(t_c)^{cp} = z_i(t_c) - (z_i(t_c) - x_i(t_c))\frac{k_2}{k} \quad (5)$$

5. Improvement 2: Linearization with Multiple Points

The usual geometric linearization [7] procedure uses two sets of points (p_{k1}, p_{i1}); (p_{k2}, p_{i2}), then the algorithm updates one point p_{ic}, in this case we use four sets of points (p_{k1}, p_{i1}); (p_{k2}, p_{i2}); (p_{k4}, p_{i4}); (p_{k5}, p_{i5}) in order to update one point p_{i3}, this leads to more precision, reliability and to reduce ambiguity.

Figure 4 shows the same scheme of Figure 2, but with five points in every line, they form triangles where it is possible to find a relationship between the cathetuses, to find the location of the unknown point pi3. To relate the points, we use the triangles formed by (δ_1,c_1), (δ_2,c_1+c_2), (δ_3,$c_1+c_2+c_3$), (d_4,$c_1+c_2+c_3+c_4$) and triangles at the bigger triangle top ($a-b-\delta_1$,c_1), ($a-c-\delta_2$,c_1+c_2), ($a-d-d_3$,$c_1+c_2+c_3$), ($a-e-d_4$,$c_1+c_2+c_3+c_4$), we do not have any information about (c_1, . . . ,c_4); because we only have PNL mixtures, we made enough equations in order to eliminate the need to use such variables. The resulting equations are shown below.

$$\frac{\delta_1}{c_1} = \frac{\delta_2}{c_1+c_2} = \frac{\delta_3}{c_1+c_2+c_3} = \frac{\delta_4}{c_1+c_2+c_3+c_4}. \quad (6)$$

$$\frac{-a+c+\delta_2}{1+\Delta} = \frac{(-a+d+\delta_3)\delta_2}{(1+\Delta)\delta_3} = \frac{-a+e+\delta_4}{(1+\Delta)(1+p+q+pq)}. \quad (7)$$

$$where \quad p = \frac{\delta_3-\delta_2}{\delta_2}, \quad q = \frac{\delta_4-\delta_3}{\delta_3}, \quad \Delta = \frac{\delta_2-\delta_1}{\delta_1}.$$

by elimination of (6) and (7)

$$c = \frac{(-a+d+\delta_3)\delta_2}{\delta_3},\ c = \frac{-a+e+\delta_4}{1+p+q+pq}. \tag{8}$$

We average the results of equation (8) to calculate c, in order to obtain a new z_i with multiple points, called z_i^{mp} to be replaced in equation (2).

$$z_i^{mp}(t_3) = x_k(t_1) + \delta_2 + c. \tag{9}$$

Table 2. Table of re between sources ar

Performance indicators	$r_{(v,\hat{v})}$		$r_{(s,y)}$		$error_{(s,y)}$		improvement %		
Method	mean	$\hat{\sigma}^2/mean$	mean	$\hat{\sigma}^2/mean$	mean	$\hat{\sigma}^2/mean$	$r_{(v,\hat{v})}$	$r_{(s,y)}$	$error_{(s,y)}$
gpICA	0.9338	0.72%	0.8574	1.78%	0.12	3.13%			
Compensation	0.9667	0.81%	0.8992	3.39%	0.1398	3.27%	3.52%	4.88%	-10.88%
Multiple points	0.9642	0.58%	0.9043	1.24%	0.0751	2.66%	3.26%	5.47%	41.24%

6. Experiment Results: Mixture of Four Speech Signals

$$A = \begin{bmatrix} -0.0870 & 0.7140 & -0.8350 & 0.4480 \\ 0.2830 & -0.5600 & -0.8420 & -0.1300 \\ -0.0860 & 0.0930 & -0.0610 & 0.5220 \\ -0.4830 & -0.7510 & 0.5310 & 0.2020 \end{bmatrix} \tag{10}$$

$$\begin{aligned} x_1(t) &= \tanh(10v_1(t)) \\ x_2(t) &= 0.1v_2(t) + v_2(t)^3 \\ x_3(t) &= \tanh(2v_3(t)) + v_3(t)^3 \\ x_4(t) &= v_4(t) + \tanh(7v_4(t)) \end{aligned} \tag{11}$$

We executed the same experiments proposed in [7] in order to evaluate the performance of our improvements to gpICA. The experiment used four speech signal with 5000 samples each (taken from [9]). The linear and non linear mixtures are given by equations (10) and (11). We use μ=0.2 in three cases. L=150 in original linearization and improvement 1. L=50 in improvement 2. We use Fast-ICA Algorithm for the blind source separation presented by Hyvarinen and Oja[4].

We performed 100 executions, the correlation coefficient, quadratic error and variation coefficient are shown in Table 2.The performance indicators were measured for the following cases (see Figure 1):

Case 1: between the linearized signal and the lineally mix sources. Case 2: between original sources and estimated signals. The variation coefficient, correlation and quadratic error were better for Improvement 2 in both cases.

The correlation was computed with (12). The variation coefficient (Standard deviation divide by mean) was used to measure data dispersion.

$$\mathbf{r}(s,y) = \frac{\sum t = 1N(s(t)-\overline{s})(y(t)-\overline{y})}{\sqrt{\sum t = 1N(s(t)-\overline{s})2\sum t = 1N(y(t)-\overline{y})2}} \tag{12}$$

The correlation mean for case 2, was greater than 0.85 for all algorithms, while for case 1 was greater than 0.93, the variation also got worse, it was less than 4%, with this information we can say that the separation process scattered the algorithms results, and declines the performance after the linearization process. Hence, the algorithms (gpICA, Improvement 1 and 2) performance does not depends on the linearization process, it also depends on the selection of a suitable linear BSS algorithm. Improvement 2 has 41% less error than original geometric linearization algorithm.

7. Conclusions

The geometric linearization algorithm with Compensation (Improvement 1), shows a new approach to find the correct position of qic (calculated point), the geometric linearization performance was increased only by adding some compensation to the point which was calculated with the usual gpICA equations, using information of the point to linearize that was not used before by the original algorithm. With the linearization with Multiple Points, performance was improved, because a larger set of points (a set of four points and a set of five points) to update every surface point position were used. This improvement is the more reliable than usual gpICA as shown by the results of experiments.

References

[1] Jean-Francois Cardoso. Blind signal separation: statistical principles. Proceedings of the IEEE, OCT. 1998.
[2] Aapo Hyvärinen and Erkki Oja. Independent component analysis: Algorithms and applications. 2000.
[3] J.F. Cardoso and Souloumiac. Blind beamforming for non-gaussian signals. 1993.
[4] A. Hyvarinen and E. Oja. A fast fixed-point algorithm for independent component analysis. Journal In Neural Computation, pages 1483,1492.
[5] Juha Karvanen, Jan Eriksson, and Visa Koivunen. Maximum likelihood estimation of ica model for wide class of source distributions. Signal Processing Laboratory.
[6] A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. IEEE Transactions on Signal Processing, 47, 1999.
[7] Thang Viet Nguyen, Jagdish Chandra Patra, and Sabu Emmanuel. gpica: A novel nonlinear ica algorithm using geometric linearization. EURASIP Journal on Advances in Signal Processing, 2007.
[8] Thang Viet Nguyen, Jagdish Chandra Patra, Amitabha Das, and Geok See Ng. Post nonlinear blind source separation by geometric linearization. Proceeding of international Conference on Neural Neural Networks, I, 2005.
[9] A Cichocky, S. Amari, K.Siwek, and all. Icalab toolbox. http://www.bsp.brain.riken.jp/ICALAB, 2003.

Authors

Erika Torres, Electronic Engineer from Distrital University of Colombia (2008). Master of control in course of University of Science and Technology of Beijing.

Paula Ulloa, Electronic Engineer from Distrital University of Colombia (2008).

Andres Gaona, Electronic Engineer from Distrital University of Colombia (2003), Master of Electronic Engineer in from Andes University of Colombia (2006), and Assistant Professor at the Faculty of Engineering at Distrital University.

Miguel Torres, Systems Engineer from National University of Colombia (1999), Master of Science in Computer Science from Mississippi State University (2003), and Assistant Professor at the Faculty of Engineering at Javeriana University.