

NONPARAMETRIC DENSITY ESTIMATION BASED INDEPENDENT COMPONENT ANALYSIS VIA PARTICLE SWARM OPTIMIZATION

D. J. Krusienski
Brain-Computer Interface Laboratory
Wadsworth Center, NYSDOH
Albany, NY

W. K. Jenkins
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA

ABSTRACT

This paper investigates the application of a modified particle swarm optimization technique to nonparametric density estimation based independent component analysis (ICA). Nonparametric ICA has the advantage over traditional ICA techniques in that its performance is not dependent upon prior assumptions about the source distributions. Particle swarm optimization (PSO) is similar to the genetic algorithm in that it utilizes a population based search suitable for optimizing multimodal error surfaces where gradient-based algorithms tend to fail, such as those generated by nonlinear entropy maximization schemes used in ICA algorithms.

1. INTRODUCTION

Independent Component Analysis (ICA) techniques have garnered much attention recently for their ability to successfully blindly separate linear mixtures of signals generated by independent sources. Although the current ICA algorithms have proven to be successful for some cases, most of the time the algorithms' performances are at best satisfactorily. The weakness of the aforementioned ICA algorithms primarily stems from two factors: the entropy estimation technique and the optimization of the entropy performance function.

The heart of ICA is a complex optimization problem of determining the unknown "unmixing" matrix. In ICA algorithms the determination of this unmixing matrix is performed based on maximizing the estimated entropy of the system. The quality of the results depends on how the entropy is estimated, which traditionally relies on apriori assumptions about the sources and mixing matrix. For most ICA algorithms, for reasonable performance, it is necessary to restrict the probability distributions of the sources to a

particular shape or class via fixed or parametric estimates [1][3]. This can lead to a simple training network, but obviously results in poor performance when the actual sources do not match the assumed distributions. Such restrictions are not placed on the sources in nonparametric density estimation approaches, making the algorithms more robust.

Regardless of whether the ICA density estimation is fixed, parametric, or nonparametric, ICA algorithms attempt to maximize the estimated entropy of the system. The preeminent ICA algorithms primarily use gradient based techniques to perform the entropy maximization. These gradient based techniques are often misguidedly deemed acceptable for several reasons. For one, because of the inherent complexity of many ICA algorithms, additional complexity is avoided by incorporating a reliable gradient based algorithm with provable (albeit often suboptimal) local convergence. Also, this suboptimal performance may occur less frequently than expected because many global optima of the performance surface exist as scaled and permuted versions of the unmixing matrix. However, for example, it is well-known and commonly overlooked that neural networks such as implemented in variations of Infomax [3][12] (and in general) are highly nonlinear and produce multimodal performance surfaces that do not lend themselves well to gradient-based techniques. Likewise other ICA techniques, including the nonparametric density estimation examined here [4], contain nonlinear entropy estimation functions. Again, gradient based optimization will inevitably lead to suboptimal solutions on such nonlinear performance functions, which requires multiple restarts to avoid a meaningless unmixing matrix. In addition, the number of local minima of the performance surface can dramatically increase when the unknown sources have multimodal distributions.

Although a global optimization alternative to a gradient based approach, such as the suggested particle

swarm optimization technique, would likely improve the performance of both parametric and neural network [13] based ICA algorithms, nonparametric density estimation based ICA is chosen for the analysis due to the aforementioned advantages.

2. NONPARAMETRIC DENSITY ESTIMATION BASED ICA

Assuming a linear mixture of N independent sources of the form $x=As$, where s is the vector of source signals and x is the matrix of mixed signals, ICA attempts to solve $y=Wx$, where W is the unmixing matrix that approximates A^{-1} to within a scaling and permutation. This is typically accomplished by maximizing the entropy, or equivalently, minimizing the mutual information between the reconstructed signals according to:

$$\min_W \left\{ - \sum_{i=1}^N H(y_i) - \log|\det W| - H(x) \right\} \quad (1)$$

Because $H(x)$, the entropy of the input, is constant with respect to the weight matrix W , it can be dropped from the expression, giving the following cost function:

$$L(W) = - \sum_{i=1}^N E[\log p_{y_i}(w_i x)] - \log|\det W| \quad (2)$$

where w_i is the i^{th} row of matrix W .

Nonparametric density estimate based ICA [4] aims to simultaneously determine the densities of the source distributions as well as the unmixing matrix. Using a batch of sample data of size M , the marginal distributions are approximated using the following:

$$p_{y_i}(w_i x^{(k)}) = \frac{1}{Mh} \sum_{m=1}^M \varphi\left(\frac{w_i(x^{(k)} - x^{(m)})}{h}\right) \quad (3)$$

Substituting in equation 2 results in the following cost function:

$$L(W) = - \frac{1}{M} \sum_{i=1}^N \sum_{k=1}^M \left[\frac{1}{Mh} \sum_{m=1}^M \varphi\left(\frac{w_i(x^{(k)} - x^{(m)})}{h}\right) \right] - \log|\det W| \quad (4)$$

where $\varphi(\cdot)$ is the Gaussian kernel, h is the kernel bandwidth, and $x^{(m)}$ is the m^{th} column of the mixture x .

The additional constraint of $\|w_i\|=1$ is also imposed to restrict the search space.

3. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization was first developed in 1995 by Eberhart and Kennedy [7], rooted on the notion of swarm intelligence of insects, birds, etc. The swarm of particles represents multiple parameter estimates, analogous to the population of individuals in the genetic algorithm. The conventional PSO algorithm begins by initializing a random swarm of R particles, each having T unknown parameters to be optimized. At each epoch, the fitness of each particle is evaluated according to the selected fitness function. The algorithm stores and progressively replaces the most fit parameters of each particle ($pbest_i$, $i=1,2,\dots,R$) as well as a single most fit particle ($gbest$) as better fit parameters are encountered. The parameters of each particle (p_i) in the swarm are updated at each epoch (n) according to the following equations:

$$\begin{aligned} \overline{vel}_i(n) &= w * \overline{vel}_i(n-1) \\ &+ acc_1 * \text{diag}[e_1, e_2, \dots, e_T]_{i1} * (gbest - p_i(n-1)) \\ &+ acc_2 * \text{diag}[e_1, e_2, \dots, e_T]_{i2} * (pbest_i - p_i(n-1)) \end{aligned} \quad (5)$$

$$p_i(n) = p_i(n-1) + \overline{vel}_i(n) \quad (6)$$

where $\overline{vel}_i(n)$ is the velocity vector of particle i , e_t are random values $\in (0,1)$, w is the inertia weight, and acc_1 and acc_2 are the acceleration coefficients toward $gbest$ and $pbest_i$, respectively.

The trajectory of each particle is influenced in a direction determined by the previous velocity and the location of $gbest$ and $pbest_i$. The two acceleration coefficients combined form what is analogous to the step size of an adaptive algorithm. The random e_t vectors provide the randomness of the step between $gbest$ and $pbest_i$. The inertia weight controls the influence of the previous velocity.

As new $gbests$ are encountered during the update process, all other particles begin to swarm toward the new $gbest$, continuing to search along the way. The search regions continue to constrict as new $pbest_i$'s are encountered. The algorithm is terminated when all of the particles in the swarm have converged to $gbest$ or a suitable minimum error condition is met.

The modified PSO (MPSO) algorithm incorporates an adaptive inertia and mutation operator that enhance the convergence properties and overall performance of

conventional PSO. The mutation operator is analogous to that of the genetic algorithm. Its purpose is to eliminate stagnation of particles that often occurs in conventional PSO. Because the mutation operator tends to slow the optimal convergence rate of PSO in general, the following adaptive inertia operator is included to compensate:

$$w_i(n) = \frac{1}{\left(1 + e^{\frac{-\Delta J_i(n)}{S}}\right)} \quad (7)$$

where $w_i(n)$ is the inertia weight of the i^{th} particle, $\Delta J_i(n)$ is the change in particle fitness between the current and last generation, and S is a constant used to adjust the transition slope based on the expected fitness range. The adaptive inertia automatically adjusts to favor directions that result in large increases in the fitness value, while suppressing directions that decrease the fitness value. The specifics of these operators, as well as the performance characteristics and other variations of PSO, are detailed in [8][9][10][11].

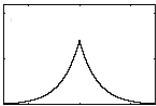
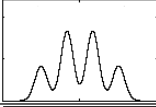
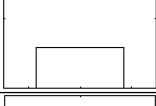
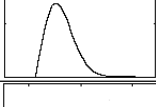
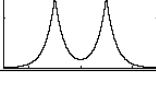
Source #	Type	Kurtosis	Pdf
1	Double exponential	3.67	
2	Gaussian mixture	-1.07	
3	Uniform	-1.21	
4	Rayleigh	0.31	
5	Double exponential mixture	-1.82	

Table 1. Synthetic source distributions used in the experiments

4. SIMULATIONS

The experimental setup consists of $N = 5$ synthetic sources drawn from different random distributions with zero mean and unit variance. The selected distributions are provided in Table 1, with zero kurtosis

corresponding to a Gaussian distribution. These sources were linearly mixed using an arbitrary mixing matrix, having condition number < 6 , generating 5 new signals. The resulting data is then whitened to aid the convergence of the algorithms. The performance results produced by the gradient based algorithm provided by the authors in [4] are compared to the results obtained by substituting the MPSO algorithm for the gradient based update of the cost function (equation 4) in the same algorithm. It should be noted that, contrary to the gradient method, no restriction on the weights is necessary with MPSO for reasonable performance. The convergence plots are the average of 50 Monte Carlo trials.

4.1 Example 1:

In this example, a data length of 500 samples from each source is used for the ICA. This small sample set is intended to test the performance at a lower extreme. A modest swarm size of 100 particles was chosen as sufficient for convergence of MPSO. The results are given in Figure 1.

4.2 Example 2:

In this example, a data length of 5000 samples from each source is used for the ICA. This sample set is intended to illustrate the case of sufficient data. Again, a swarm size of 100 particles was selected. The results are given in Figure 2.

5. DISCUSSION

The analysis presented in this paper focuses on the convergence properties of the respective optimization techniques for the specific ICA cost function given by equation 4. For the examples considered, it was verified that the minimization of equation 4 directly resulted in a lower Amari index [1], which characterizes the desired matrix composition and separation performance.

The results in Figures 1 and 2 indicate that the performance of gradient method is suboptimal compared to MPSO. For individual trials in general, the gradient method curiously tends to exhibit periods of very slow convergence, followed by periods of more rapid convergence - an effect that is averaged out but still apparent in the convergence plots. This is believed to be due to the traversing of regions having minimal gradients on the nonlinear surface. This behavior seems to be a precursor for increasingly poor performance

under progressively more extreme conditions such as more complex multimodal distributions or larger scale mixtures, for instance. The suboptimal convergence is not as noticeable, by comparison, in the larger data case. This is likely due to the fact that the performance surface is better defined and more amenable to a gradient algorithm when more data is available.

Conversely, MPSO and stochastic search algorithms are more robust with more consistent convergence characteristics. This is because the performance surface gradients do not explicitly affect the weight updates. Thus, stochastic algorithms can be applied to wide-ranging nonlinear optimization problems with reliable performance. In addition, when compared to number of operations necessary for some complex gradient based weight updated schemes, MPSO and similar algorithms do not add a considerable computational burden [11]. For these reasons, it is practical to consider stochastic optimization algorithms, not only as a viable alternative, but as a replacement to conventional ICA optimization techniques - especially in instances where the conventional techniques falter.

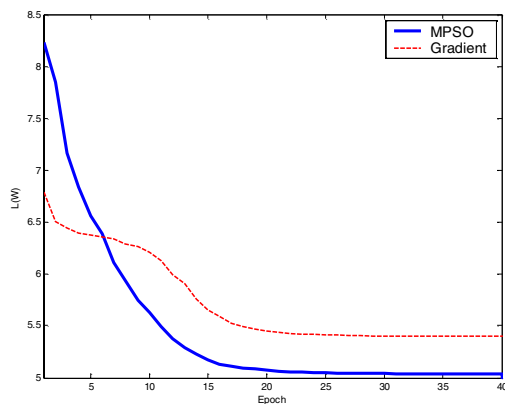


Fig. 1. Learning curves for Example 1

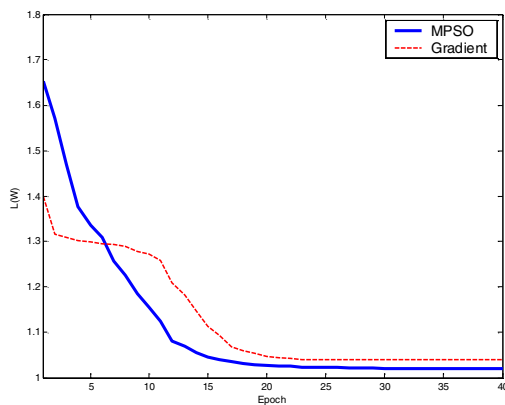


Fig. 2. Learning curves for Example 2

6. REFERENCES

- [1] Amari, S., Cichoki, A., and Yang, H.H., "A new learning algorithm for blind signal separation," *Advances in Neural Information Processing Systems*, vol.8, MIT Press, 1996, pp. 757-763.
- [2] Bach, F.R. and Jordan, M.I., "Kernel Independent Component Analysis," *J. Machine Learning Res.*, vol.3, 2002, pp. 1-48.
- [3] Bell, A.J., Sejnowski, T.J., "An Information Maximization Approach to Blind Separation and Blind Deconvolution". *Neural Computation*, 7, 6, 1995, pp. 1129-1159.
- [4] Boscolo, R., Pan, H., Roychowdhury, V.P., "Independent Component Analysis Based on Nonparametric Density Estimation," *IEEE Transactions on Neural Networks*, Vol. 15, No. 1, January 2004.
- [5] El-Gallad, A. I., El-Hawary, M. E., Sallam, A. A., and Kalas, A. "Enhancing the particle swarm optimizer via proper parameters selection," *Canadian Conference on Electrical and Computer Engineering*, 2002, pp. 792-797, 2002.
- [6] Hyvarinen, A., "Survey on Independent Component Analysis," *Neural Comp. Surveys*, vol.2, 1999, pp. 94-128.
- [7] Kennedy, J., Eberhart, R. C., and Shi, Y., *Swarm Intelligence* San Francisco: Morgan Kaufmann Publishers, 2001.
- [8] Krusienski, D. J. and Jenkins, W.K., "Design and Performance of Adaptive Systems Based on Structured Stochastic Optimization Strategies", *IEEE Circuits and Systems Magazine*. (to appear)
- [9] Krusienski D. J. and Jenkins, W.K., "A Particle Swarm Optimization-LMS Hybrid Algorithm for Adaptive Filtering", *Proc. of the 38th Asilomar Conf. on Signals, Systems, and Computers*, November 2004. (to appear)
- [10] Krusienski, D. J. and Jenkins, W.K., "Particle Swarm Optimization for Adaptive IIR Filter Structures," *Proc. of the 2004 Congress on Evolutionary Computation*, June 2004.
- [11] Krusienski, D. J., "Enhanced Structured Stochastic Global Optimization Algorithms for IIR and Nonlinear Adaptive Filtering", Ph.D. Thesis, The Pennsylvania State University, 2004.
- [12] Lee, T.W., Girolami, M., and Sejnowski, T.J., "Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources," *Neural Computation*, vol.11, no.2, 1997, pp. 417-441.
- [13] Mendes, R., Cortez, P., Rocha, M., Neves, J., "Particle swarms for feedforward neural network training," in *Proc. Int. Joint Conf. Neural Networks (IJCNN '02)*, vol. 2, 2002, pp. 1895-1899.