Using Fair M-Estimators as a cost function for FASTICA

Reedip Banerjee
Hydrator Division
NEC Technologies India Limited
Noida, Uttar Pradesh, India
reedip.banerjee@nectechnologies.in

***Abstract*— Independent Component Analysis (ICA), useful for Blind Source Separation, is a method of decomposing an observed dataset, which has been created by the influence of multiple sources, into the set of statistically independent variables which created it. ICA uses cost function to process the dataset, and it requires that the cost function used for separation be robust, consistent and non-quadratic in nature to properly differentiate the individual components. It is also required that the cost function is computationally simple, converges quickly and does not fail to converge when applied on different data sets .One of the ideal methods for Independent Component Analysis is FastICA. Given the apparent freedom to choose non-linearites, it is proposed here to use Fair M-Estimator as a cost function for the FastICA. M-Estimators are a generalized case of Maximum Likelihood Estimators. Huber proposed M-Estimators for estimating the likelihood of a variable contained in a normal distribution, which has been effected by outliers. The algorithm obtained from this cost function is simple to implement, and is useful in removing outliers from an observed dataset. Simulations are run to compare the algorithm on non-gaussian and real life speech examples against standard FastICA cost functions. The separating capability, along with convergence speed and the ability to converge successfully is observed in this paper.**

Keywords*— *Fast ICA,Negentropy,M-Estimators

I. INTRODUCTION

Blind Source Separation is used to estimate the actual source signals, whose mixing created an observed set of signals. Independent Component Analysis is possibly the most common and widely known algorithm used for Blind Source Separation. Independent Component Analysis (ICA) decomposes an observed multi-variable set and attempts to estimate the set of statistically independent variables which created it.

The ICA model assumes that a linear mixture of some unknown source signals, in an unknown mixing system have created the observed variables. The observed data might have also been affected by external noise, which can be gaussian in nature. However, for proper estimation, source signals are expected to be non-gaussian, and mutually independent though they may not be in the real world scenario. The source signals are known as Independent Components which can be determined by ICA [1].The only exception to this fact is that ICA allows maximum of one Gaussian signal to exist.

A. ICA

Assume the existence of an observed**,** multidimensional signal '**x**' which can be described as $x = [x_1, x_2, x_3, x_4 \ldots x_n]^T$. The observed signal is the result of mixing various source signals, each of which is assumed mutually independent, which means that the change on one signal does not have an impact on the other signals. The signal 'x' can hence be represented as a vector which has been created by the mixing matrix '**A**' operating on the original independent signals '**s**':

$$\mathbf{x(t)= A.s(t) +} \partial \qquad (1)$$

∂ in (1) is an additional noise which may be present during the recording of observed data. ICA estimates the demixing matrix '**w**' to recover the signal. If **y** is the set of estimated signals recovered by executing ICA on the observed signals, then

$$\mathbf{y(t)=w.x(t)} \qquad (2)$$

B. FastICA

FastICA, a derivative of Independent Component Analysis is a computationally efficient algorithm. The algorithm attempts to maximize the non-gaussianity between the signals as a measure of statistical independence [1, 6].The FastICA has several of the advantages of neural algorithms: It can determine all signals in parallel, it is distributed, computationally simple, and requires little memory space [4].

Non-Gaussianity can be measured using Kurtosis and Negentropy. Kurtosis, however, is sensitive to outliers, which is reduces the robustness of the algorithm. Therefore, FastICA algorithm iteratively maximizes an approximate of the *negentropy* of the observed data. Since, among all variables of equal variance, the largest entropy is observed in gaussian variables; negentropy can be used to define a measure of nongaussianity [3].As per [2], ICA and therefore FastICA as its derivative, acts to maximize Negentropy, using the equation,

$$\sum_{i=1}^{n}[\mathrm{E}\{\mathrm{G}(\mathrm{w}_i^{\mathrm{T}}\mathrm{x})\} - \mathrm{E}\{\mathrm{G}(\mathrm{f})\}]^2 \quad (3)$$

with the constraint that $E\{(w^T x)^2\}=1$, where 'G' is considered to be almost any non-quadratic function, 'f' is a Gaussian random variable of unit-variance and 'x' is an n-variable vector. Convergence means that the old and new values of **w** point in the same direction, i.e. their dot-product are nearly equal to 1, maximizing negentropy. It is not necessary that the vector converges to a single point [5].

The basic form of the FastICA algorithm is as follows:

- Choose an initial weight vector **w**.
- Let $\mathbf{w}^+_i = E\{\mathbf{z_i(t)}g(\mathbf{w}^T\mathbf{z_i(t)})\} - E\{g(\mathbf{w}^T\mathbf{z_i(t)})\}\mathbf{w}$
- Let $\mathbf{w_{i+1}} = \mathbf{w}^+_i/\sqrt{(\mathbf{w}^{+T}_i)(\mathbf{w}^+_i)}$
- If not converged, go back to 2.

Where $\mathbf{z_i(t)}$ is the pre-whitened observed data **x**, at iteration *i* and time ***t***. Pre-whitening involves a linear transformation, to create uncorrelated entries from the observed signals. As FastICA is known to be sensitive to its initialization, therefore in the below study the initial weight has been kept constant for all the studied samples. FastICA reinitializes the weight if it doesn't converge. One of the greatest advantages of FastICA algorithm is that it finds independent components of almost any non-Gaussian distribution using any non-linearity ***g*** [1,8]. However, like other similar algorithms, the selection of non-linearity **g** may depend on the probability distribution function (p.d.f) of the original source signal. The only condition for non-linearity **g** is that it should be a non-quadratic function in nature [7].The performance of FastICA can be optimized by choosing a suitable nonlinearity *g* for specific distributions. However, non-linearites suggested in [7] can cover most signal distributions, though no proof of the same has been given.

II. M-ESTIMATORS AS COST FUNCTION FOR FASTICA

A. Introduction to M-Estimators

M-Estimators are a generalized case of Maximum Likelihood Estimators, proposed by Huber. It was proposed for estimating the likelihood of a variable contained in a normal distribution, which has been effected by outliers. Therefore, given a set of observed data, they are used to estimate the p.d.f which would most likely result in the actual source [6].Unlike Least Square Method; M-Estimators utilize a cost function ρ (k) to reduce the effect of outliers, thus making it more robust in nature and unlike the Maximum Likelihood Estimation method, M-Estimator does not depend on any model to estimate the p.d.f. It is the shape of ρ (k) which controls the accuracy and robustness of the estimated value. This is because the knowledge about the actual signal is not known. The derivative Ψ(x) = d ρ (x)/dx, which is also called the influence function, measures the influence of an observed variable on the value of the estimate.

If the observed variables are a set *a(k)*, with $1 \le k \le M$, then M-Estimators are used to estimate the actual signal a*, where

$$a^* = \arg(\min_{a^*}\{\sum_{i=0}^{m}\rho(a,a^*)\}) \quad (4)$$

M-Estimators need to minimize (3), and therefore

$$\sum_{i=0}^{M}\Psi(a,a^*) = 0 \quad (5)$$

B. Application of Fair M-Estimators for FastICA

FastICA supports almost any non-quadratic function as it cost function. Also, M-Estimators are very robust against outliers, which are a required property of cost functions for FastICA, as specified in [2] although this particular property is not the focus of this paper. It is also to be mentioned that FastICA uses Maximum Likelihood Estimation as specified in [1], which motivated us to use M-Estimators. Therefore, in the current study, the Fair M-Estimator, defined in Table I, has been considered as a cost function for FastICA.

TABLE I: M-Estimator ρ(x) and Influence Function Ψ(x)

	ρ(x)	Ψ(x)
Fair M-Estimator	$A^2\left(\frac{\lvert x\rvert}{A} - \ln\left(1+\frac{\lvert x\rvert}{A}\right)\right)$	$\frac{x}{\left(1+\frac{\lvert x\rvert}{A}\right)}$

Fair M-Estimator was defined by Rey (1983). It has defined continuous derivatives of first 3 orders, and unlike other M-Estimators like Cauchy and Welsch, it yields a unique solution. The value 'A' is a Cramer-Rao based tuning parameter, used for trading off high efficiency with robustness. It has been found that the tuning constant A has 95% asymptotic efficiency at 1.3998 [9], though it is subject to p.d.f of the source signals, similar to other cost functions of FastICA like tanh and pow3. The same tuning parameter has been used in the below evaluations, as the tuning parameter did not have any effect on the Amari performance, as shown in the simulations below.

III. SIMULATION

The FastICA algorithm has been evaluated, using the Fair M-Estimator as the cost function, and 3 zero-mean non-gaussian signals, given in Figure I on MATLAB. The execution had 100 iterations, on signal samples ranging from 500-5000. The convergence of the Fair M-Estimator function has also been evaluated for the first 5speech signals, obtained from http://research.ics.tkk.fi/ica/cocktail/cocktail_en.cgi, which have 50000 samples per signal .The Fair M-Estimator was also compared for both the scenarios with the 2 'original' non-linearites, i.e. tanh and pow3. The mixing matrix, A, was randomized, as could be the case in real world scenarios. While the evaluation result for the 1st case is shown in Figure 2, the evaluation result of the 2nd case is shown in Table 2.

As specified in [7], there may not be a single tuning constant value which could be used to easily separate the observed signals for the M-Estimators. Figure 3 shows the average number of iterations of Fair M-Estimator as its tuning parameter is modified, for sample size of 100-500, using the non-zero unit variance signal mixture. Table 3 shows the performance observed during the above evaluation. Please note that an additional gaussian noise, ranging from -20dB to +20dB was added in all the scenarios, and the results are an average of the execution.

Fig. 1. *Comparison of log Amari Performance: Fair vs. tanh vs. pow3*

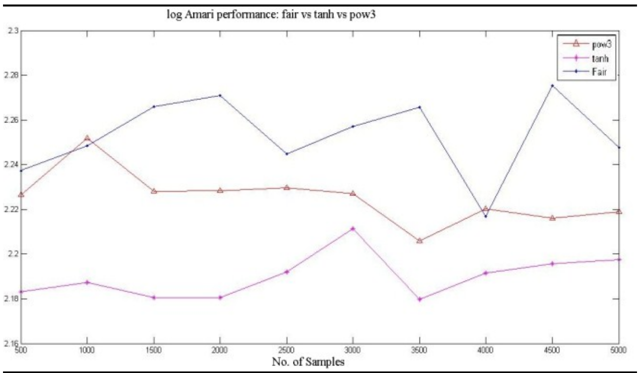
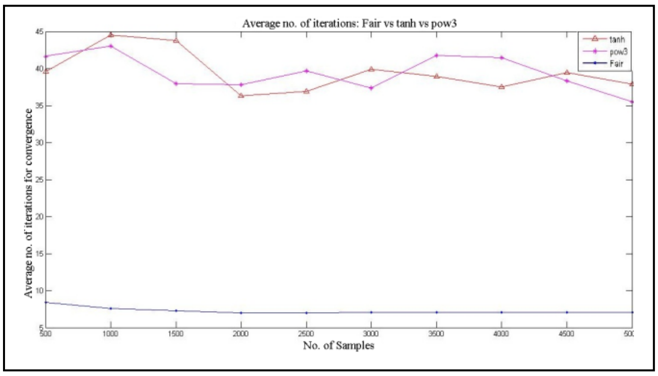


Fig. 2. *Comparison of Average No. of Iterations for convergence: Fair vs. tanh vs. pow3*



As observed from Figure 2, the Separation Performance of Fair M-Estimator may not be good as compared to tanh and pow3, however, the average number of iterations required for reaching convergence is very low. This is an important feature as FastICA algorithms require the processing to be fast. Separation Performance may be improved by better pre-processing of the data.

TABLE II: Evaluation of Cost Functions for Speech Signals

Cost Function	Iterations for Convergence	Log Amari Performance	Failure to Convergence
Fair	4.3863	-1.0160dB	0
pow3	4.4668	-1.8752dB	197
tanh	4.3415	-1.9501dB	205

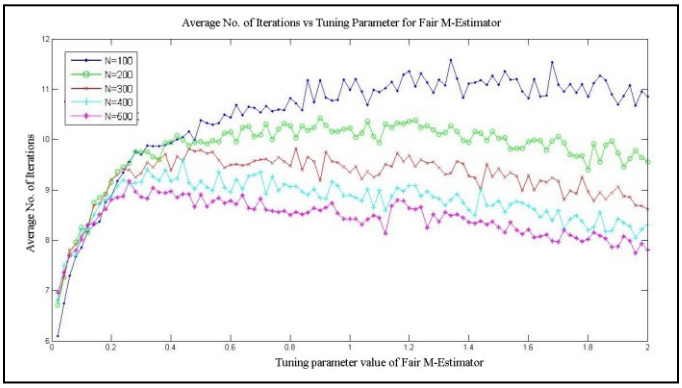
The most important fact to be noted in Table 2, which was also noted while performing other evaluations, was *that Fair M-Estimator never failed to converge.*

TABLE III: Average performance comparison for Fair M-Estimator

Tuning Parameter	Min Performance	Max Performance
0.02	2.2264	2.2712
0.2	2.2394	2.2884
0.4	2.2430	2.2745
0.6	2.2199	2.2771
0.8	2.2404	2.2730
1.0	2.2453	2.2819
1.2	2.2391	2.2899
1.3998	2.2368	2.2706
1.6	2.2414	2.2766
1.8	2.2370	2.2807
2.0	2.2298	2.2796

Failure to Converge meant if the FastICA algorithm could not converge in the maximum defined iterations, which in the current case was 1000.Tanh and pow3 sometimes did not converge in the maximum defined iterations, but Fair M-Estimator was always converging which was not expected, as convergence depends not only on the p.d.f of the signal but also on the initialization of the weight matrix, and the randomness by which the data was mixed.

Fig. 3. *Average Number of Iterations vs. Tuning Parameter for Fair M-Estimator*



It is possible that the initialization of weight matrix, which has been considered as Identity matrix for all the executions may have played a role, but it also suggests that if the right initialization matrix is chosen, FastICA could be allowed to converge for almost any signal using the Fair M-Estimator. As can be observed from Table 3, there is not much impact of the tuning parameter on the performance of the M-Estimator. The performance values lie within a specific range. However, the number of iterations may be affected by the tuning constant, as demonstrated in Figure 3, with the average iterations increasing as the tuning parameter is increased.

IV. CONCLUSION

FastICA depends not only on the cost function to compute the independent signals; it also takes into consideration the p.d.f of the source, as mentioned in [2]. Since any non-quadratic function can be used for FastICA, therefore, it is proposed in this paper to use Fair M-Estimator, and its performance for different signals was compared, along with tanh and pow3 cost functions. The evaluations show that the tuning parameter of Fair M-Estimator does not adversely affect its performance, though it affects the number of iterations taken for convergence. Fair M-Estimator may not perform as well as tanh and pow3 on occasions, but it take, on an average, very few steps for convergence and net performance of the algorithm can be improved further by better pre-processing. It was also found in different executions, for different signal sources that if the 3 cost functions were evaluated on the same set of signals, randomly mixed and effected by a Gaussian noise, Fair M-Estimator never failed to converge in the analysis, though tanh and pow3 did.

V. REFERENCES

[1] Hyvarinen A. “Independent Component Analysis: Algorithms and Applications”. Neural Networks, 2000, 13:411-43.

[2] Hyvarinen A.,”Fast and Robust Fixed-Point Algorithms for Independent Component Analysis”, Neural Networks, April 23,1999.

[3] Hyvarinen, A., Karhunen, J., Oja, E., “Independent Component Analysis”, John Wiley & Sons, New York (2001).

[4] Scott C. Douglas, Malay Gupta,Hiroshi Sawada, and Shoji Makino“Spatio–Temporal FastICA Algorithms for the Blind Separation of Convolutive Mixtures “ IEEE transactions on Audio,Speech,and Language Processing, Vol. 15, No.5, July 2007 .

[5] Malaya K. Nath,” Independent Component Analysis of Real Data “, 2009 Seventh International Conference on Advances in Pattern Recognition.

[6] In Jae Myung,”Tutorial on Maximum Likelihood Estimation”,Journal of Mathematical Psychology, October 2002.

[7] Chao,Jih-Cheng and Douglas,Scott C.,”A Simple and Robust FastICA Algorithm using the Huber M-Estimator Cost Function”,ICASSP,2006.

[8] Zahooruddin,Farooq Alam Orakzai,"Hardware implementation of blind source separation of speech signals using independent component analysis",International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10 No: 01

[9] Zhengyou,Zhang, Parameter Estimation Techniques,”A Tutorial with Application to Conic Fitting”,Image and Vision Computing, Vol.15, No.1, pages 59-76, January 1997.

[10] Dutter,R. , Filzmoser, P. ,Gather,U.,Rousseeuw,P.,” Developments in Robust Statistics”,Physica-Verlag,2003