A Comparative Analysis of M-Estimators as a Cost Function for FASTICA

Reedip Banerjee

University School of Information Technology, Guru Gobind Singh Indraprastha University

Delhi, Delhi, India

reedipbanerjee@gmail.com

Udayan Ghose

University School of Information Technology, Guru Gobind Singh Indraprastha University

Delhi, Delhi, India

g\_udayan@lycos.com

Abstract— **Independent Component Analysis requires that the cost function used for separation be robust, consistent and non-quadratic in nature. Given the apparent freedom to choose non-linearites, we propose here to use Cauchy, Huber, Welsch and Geman-McLure M-Estimators as individual cost functions for FastICA. Algorithms obtained from these cost functions are simple to implement. Simulations are run to compare the algorithms 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**

Keywords: M- Estimators, FastICA, Negentropy

# **Introduction**

Given a set of observed signals, we use Blind Source Separation to estimate the actual source signals, whose mixing created the observed set. Blind Source Separation (BSS) has advanced a lot in the past few years with regards to improvising and creating efficient algorithms to application in various fields. Among the various algorithms proposed for BSS, Independent Component Analysis is possibly the most common and known algorithm. Independent Component Analysis (ICA) is a method of decomposing a multiple variable dataset into the set of statistically independent variables which created it.

ICA defines a model for the observed data in which, the observed variables are assumed to be linear mixtures of some unknown source signals, with an unknown mixing system. The source signals are assumed non-gaussian, and mutually independent and they are called the independent components of the observed data. These independent components can be found by ICA [1].However, it is possible for ICA to estimate all the independent components even if one of the source signals is Gaussian in nature.

## ICA

Let us assume that we have an observed signal ‘**x’.** It isa multidimensional variable, and can be described as x = [x1,x2,x3,x4……xn]T. The observed signal is itself a collection of various source signals, each of which is assumed mutually independent, meaning there is no impact on the other signals if one of them experiences any changes. We can hence write the signal ‘x’ as a vector which has been created by the mixing matrix ‘**A’** operating on the original independent signals ‘**s’**:

**x(t) = As(t) + Ω** (1)

**Ω** in (1) is the additional noise present during the recording of observed data. To recover the signal, we estimate the demixing weight matrix ‘**w**’. If **y** is the set of estimated signals recovered by executing ICA on the observed signals, then

**y(t)=wx(t)** (2)

## FastICA

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

Non-Gaussianity can be measured using Kurtosis and Negentropy. However, due to the fact that Kurtosis is not robust to effect of outliers, FastICA algorithm iteratively maximizes an approximation of the negentropy of the observed data. Since, among all variables of equal variance, gaussian variables have the largest entropy; negentropy can be used to define a measure of nongaussianity [3].As per [2], ICA in itself, and FastICA as its derivative, acts to maximize the approximation of negentropy, using the equation,

**{G(wiTx)}-E{G(f)}]2** (3)

with the constraint that E{(wTx)2}=1, where ‘G’ can be almost any non-quadratic function, ‘f’ a unit-variance Gaussian random variable and ‘x’ 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 [5].

The basic form of the FastICA algorithm is as follows:

* Choose an initial weight vector **w**.
* Let **w**+i= *E*{**zi(t)***g*(**w***T***zi(t)**)}－*E*{*g*(**w***T***zi(t)**)}**w**
* Let **wi+1** = **w**+i/**√(w+Ti)(w+i)**
* If not converged, go back to 2.

where **zi(t)**  is the estimated source at time *t* and iteration *i*. FastICA assumes that the data, **x,** has been prewhitened to zi(t)using a linear transformation, to create uncorrelated entries. Since FastICA is known to be sensitive to its initialization, therefore in our study we have kept the initial weight constant for all the studied samples. FastICA reinitializes the weight if it doesn’t converge. The algorithm 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], as mentioned earlier. The performance of FastICA can be optimized by choosing a suitable nonlinearity ***g*** for specific distributions. Non-linearites suggested in [7] can cover most signal distributions, though no proof of the same has been given.

# **M-Estimators as Cost Function for FastICA**

## 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. 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 is called the influence function. Ψ (x) measures the influence of an observed variable on the value of the estimate.

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

**a\*=arg min a\* (a,a\*)**(4)

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

**Ψ (a,a\*) =0**  (5)

## Application of M-Estimators for FastICA

The idea of FastICA supporting almost any cost function for its algorithm motivated us to experiment with the set of M-estimator functions as the cost function for FastICA, along with the fact that M-Estimators are very robust against outliers, which is a required property of cost functions for FastICA, as specified in [2].In our study, we consider four M-Estimators, defined in Table I as cost function for FastICA [9]

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

| M-Estimator | ρ(x) | Ψ(x) |
| --- | --- | --- |
| Huber | * x2/2 , |x| <= khuber * khuber (|x| - khuber /2) , |x| > khuber | * x , |x| <=khuber * khuber (sign(x)) , |x| > khuber |
| Cauchy | ck2 (log(1+(x/ck)2)/2 | x/(1+(x/ck)2) |
| Welsch | c2 (1 - exp(x/c)2)/2 | x (exp-(x/c)2) |
| Geman-McLure | x2 /2 (1+x2) | x /(1+x2) 2 |

The values ‘khuber ’, ‘ck ‘ and ‘c’ acts as a tuning constant over which the estimation occurs. Huber’s function has a linear shape above the tuning constant ‘khuber’, but has a parabolic shape below ‘khuber’, in the vicinity of zero. These functions can considerably reduce or even eliminate the influence of outliers. The advantage of using the influence function of M- Estimators for FastICA algorithm is that the algorithm becomes simple, using only multiplication, addition and relational operations. Also, Huber’s M-Estimator has already been studied in [7] and seems to be a likely candidate as a cost function. Welsch’s M-Estimator is very similar to the practical cost function given in [2], and therefore has been chosen as a candidate as well.

# **Simulation**

We evaluated the FastICA algorithm, using the M-Estimators as its cost function and running it for 100 iterations to determine a consistent average, using 3 zero-mean non-gaussian signals to create the observed samples, as given in Figure I, on MATLAB. The observed mixture had sample size N ranging from500-5000, with each sample space separated by 500samples. The observations for the above execution are demonstrated in Figure II. We also compared the performance of the M-Estimator based FastICA algorithm with the 2 ‘original’ non-linearites, i.e. tanh and pow3, on speech signals, as real life samples were not studied in [7].The speech signals were obtained from http://research.ics.tkk.fi/ica/cocktail/cocktail\_en.cgi. The observations of the speech signals have been reported in Table II. The mixing matrix, A, was randomized for both the cases defined above. An additional gaussian noise, ranging from -20dB to +20dB was added to the observed samples.

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. As per our analysis of the different signals, we found that Cauchy’s M-Estimator converged when ckε (0.06,0.17)U(0.225,2.05)while Welsch converged when cw ε (0.23,0.3)U(1.9,1.15). Huber converged for all values of khuber ε(0,2] Please do note that the performance of all three M-Estimators, i.e. Cauchy, Welsch and Huber, depended heavily on the mixture of the Gaussian noise, along with the p.d.f of the observed signals. This was determined experimentally by executing the M-Estimators on the speech signal, as well as on the zero – mean signal, for the same initial weight matrix and gaussian noise, but for different values of tuning parameters. It was found that *Cauchy, Welsch and Huber M-Estimator’s tuning parameter does not have an impact on the number of iterations required for convergence,* when they were executed on the same signal sets for a range of positive values of *ck, cw and khuber , with* ck ranging from 0 to 3,cw ranging from 0 to 3 and khuber ranging from 0 to 2.

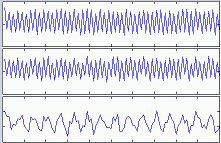
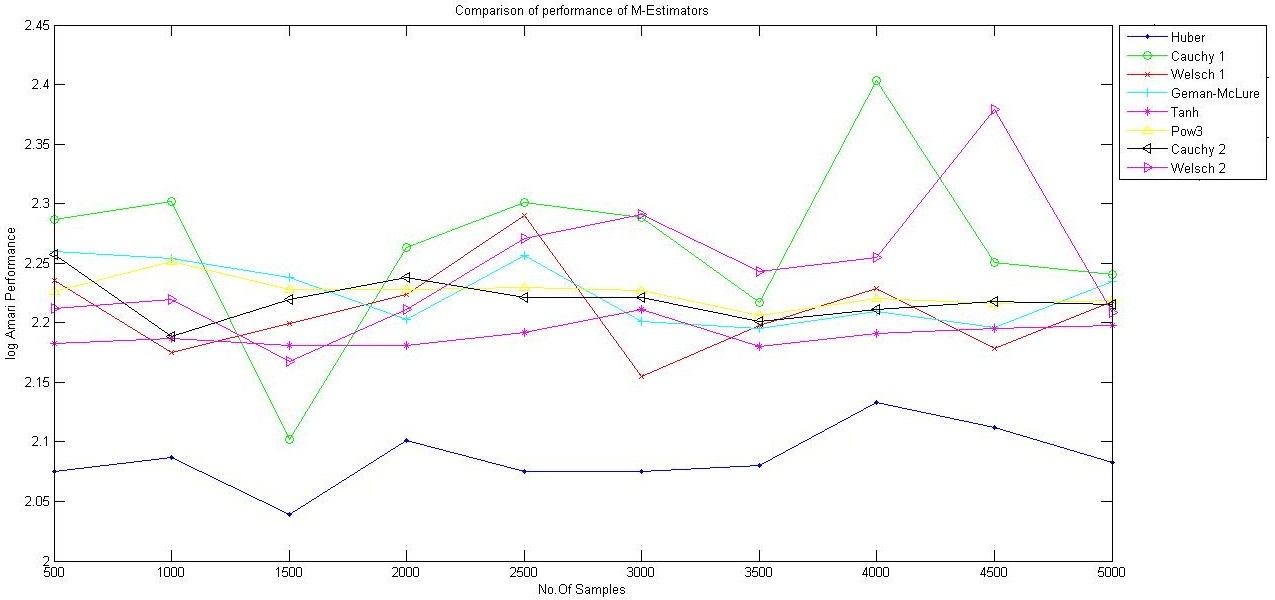


Figure I. Signals used for analysis



…..

Figure II.(a) Comparison of log Amari Performance of M-Estimators

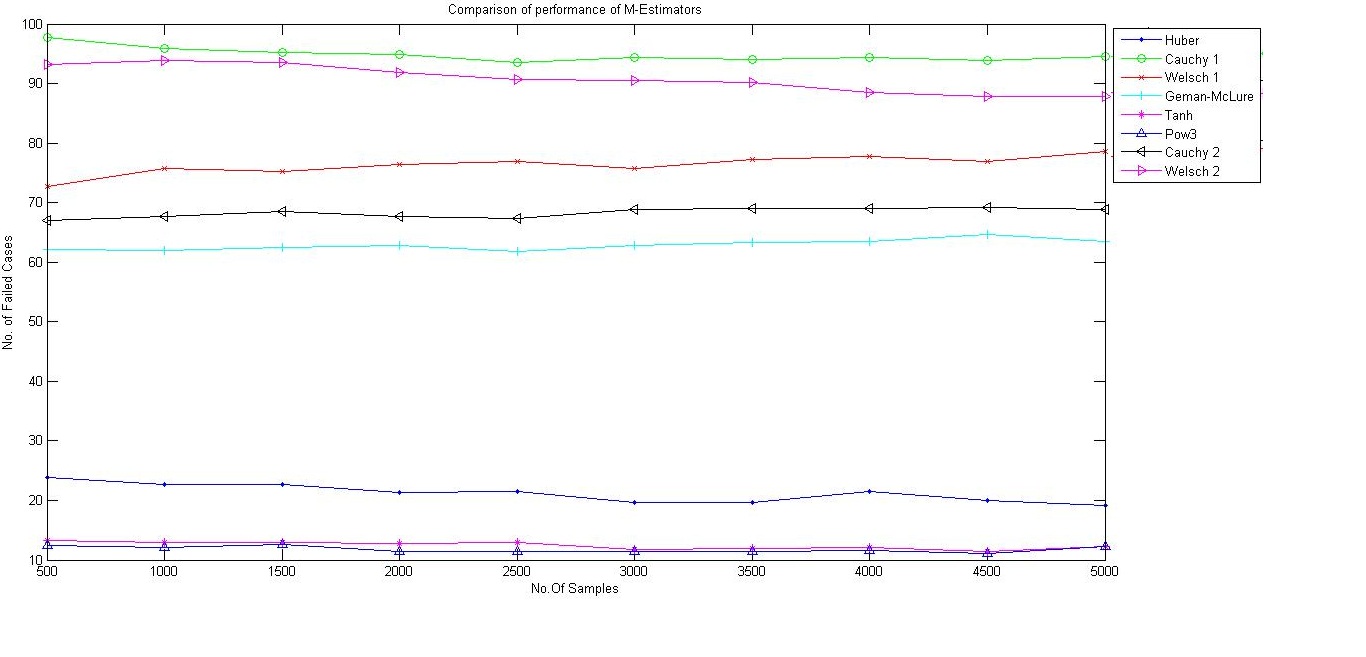


Figure II (b) Comparison of Failure to Convergence for M-Estimators

As can be determined from Figure II.(a), the performance of Huber M-Estimator is much better than other M-Estimators.

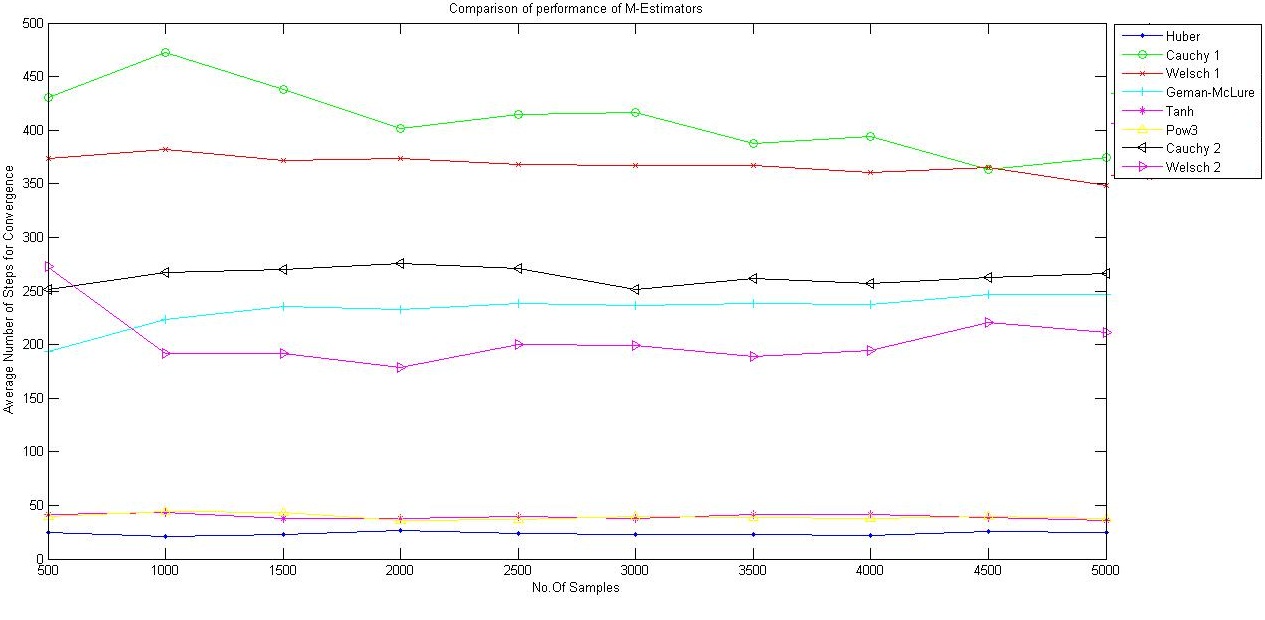


Figure II (c) Comparison of Average Steps for Convergence of M-Estimators

The tanh and pow3 cost functions were observed to be stable as per log Amari performance as they deviated very meagerly. Performance of Cauchy M-Estimator for ck ε (0.06,0.17) was the worst with maximum deviation, followed by Welsch with cw ε (0.23, 0.3). As per Figure II.(b) we could see that Huber performed exceptionally well and failed to converge just less than 25% times on an average., though tanh and pow3 were better in terms of convergence . Cauchy and Welsch, for ck ε (0.06,0.17) and cw ε (0.23, 0.3), failed nearly 90% times while other M-Estimators performed better. Failure to Convergence is observed if FastICA algorithm does not converge in maximum number of iterations, which in our case was 1000. As per Figure II.(c), Huber had the lowest number of iterations for reaching convergence, better than tanh and pow3.Cauchy M-Estimator for ck ε (0.06,0.17) performed the worst as observed.

The Table II shows the behavior of the M-Estimators when treated with real life signals with mixed with the gaussian noise. Different numbers of signals were mixed together with a random mixing matrix and an average of 100 iterations was used for the below data. As per the output obtained, we could see that Cauchy, Welsch and Geman-McLure’s Failure to Convergence rate, and the number of steps needed for Convergence was drastically affected as the number of components increased. Huber’s M-Estimator performed very well, better than other M-Estimators and sometimes better than even tanh and pow3.

1. Comparison of various non-linearites in fastica for speech signals

| No. of mixed Source Signal | Huber | C1 | W1 | GM | Tanh | pow3 | C2 | W2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N,F | N,F | N,F | N,F | N,F | N,F | N,F | N,F |
| 2 | 0.030,0 | 1.864,49 | 6.034,14 | 2.510,0 | 0.030,0 | 0.030,0 | 2.415,1 | 0.888,2 |
| 3 | 0.041,0 | 0.960,59 | 2.707,39 | 1.699,1 | 0.041,0 | 0.051,0 | 0.901,9 | 0.961,5 |
| 4 | 0.040,0 | 4.591,83 | 4.476,70 | 3.552,16 | 0.040,0 | 0.050,0 | 3.817,24 | 3.957,16 |
| 5 | 0.061,0 | 7.748,98 | 10.469,98 | 11.354,82 | 0.040,0 | 0.050,0 | 9.556,80 | 6.446,63 |
| 6 | 0.051,0 | 0.00,100 | 0.000,100 | 9.714,91 | 0.061,0 | 0.071,0 | 3.073,92 | 7.717,70 |
| 7 | 0.081,0 | 0.00,100 | 0.00,100 | 11.332,93 | 0.051,0 | 0.061,0 | 30.907,96 | 8.460,80 |
| 8 | 0.051,0 | 0.00,100 | 0.00,100 | 0.00,100 | 0.051,0 | 0.071,0 | 61.250,99 | 48.525,94 |

# **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, in this paper, we were motivated to use Cauchy, Welsch, Huber and Geman-McLure M-Estimator as FastICA cost functions. Their performance for different signals was compared with tanh and pow3 cost functions. It was determined that all M-Estimators are sensitive to the p.d.f of the source, and while Cauchy, Welsch and Geman-McLure may not be very effective, Huber performed exceptionally well in most situations. Additionally it was also found that the tuning constants do not affect Cauchy, Welsch and Huber, as the number of iterations for convergence is almost the same for the same observed signal set.

# **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.