Zhijian Yuan and Erkki Oja. A FastICA Algorithm for Non-negative Independent Component Analysis. *In Puntonet, Carlos G.; Prieto, Alberto (Eds.), Proceedings of the Fifth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2004)*, Springer Lecture Notes in Computer Science 3195, pp. 1-8, Granada, Spain, 2004.

A FastICA Algorithm for Non-negative Independent Component Analysis

Zhijian Yuan and Erkki Oja*

Helsinki University of Technology
Neural Networks Research Centre
P.O.Box 5400, 02015 HUT, Finland
{zhijian.yuan,erkki.oja}@hut.fi

Abstract. The non-negative ICA problem is here defined by the constraint that the sources are non-negative with probability one. This case occurs in many practical applications like spectral or image analysis. It has then been shown by [10] that there is a straightforward way to find the sources: if one whitens the non-zero-mean observations and makes a rotation to positive factors, then these must be the original sources. A fast algorithm, resembling the FastICA method, is suggested here, rigorously analyzed, and experimented with in a simple image separation example.

1 The Non-negative ICA Problem

The basic linear instantaneous ICA mixing model $\mathbf{x} = \mathbf{A}\mathbf{s}$ can be considered to be solved, with a multitude of practical algorithms and software; for reviews, see [1, 3]. However, if one makes some further assumptions which restrict or extend the model, then there is still ground for new analysis and solution methods. One such assumption is *positivity or non-negativity* of the sources and perhaps the mixing coefficients; for applications, see [9, 5, 13, 2]. Such a constraint is usually called *positive matrix factorization* [8] or *non-negative matrix factorization* [4]. We refer to the combination of non-negativity and independence assumptions on the sources as *non-negative independent component analysis*.

Recently, Plumbley [10, 11] considered the non-negativity assumption on the sources and introduced an alternative way of approaching the ICA problem, as follows. He calls a source s_i *non-negative* if $\Pr(s_i < 0) = 0$, and *well-grounded* if $\Pr(s_i < \delta) > 0$ for any $\delta > 0$; i.e. s_i has non-zero pdf all the way down to zero. The following key result was proven [10]:

Theorem 1. Suppose that $\mathbf{s}$ is a vector of non-negative well-grounded independent unit-variance sources $s_i,\ i = 1, ..., n$, and $\mathbf{y} = \mathbf{Q}\mathbf{s}$ where $\mathbf{Q}$ is a square orthonormal rotation, i.e. $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. Then $\mathbf{Q}$ is a permutation matrix, i.e. the elements y_j of $\mathbf{y}$ are a permutation of the sources s_i, if and only if all y_j are non-negative.

* This work was supported by the Academy of Finland as part of its Center of Excellence project "New Information Processing Principles".

C.G. Puntonet and A. Prieto (Eds.): ICA 2004, LNCS 3195, pp. 1–8, 2004.

The result of Theorem 1 can be used for a simple solution of the non-negative ICA problem. The sources of course are unknown, and $\mathbf{Q}$ cannot be found directly. However, it is a simple fact that an arbitrary rotation of $\mathbf{s}$ can also be expressed as a rotation of a pre-whitened observation vector. Denote it by $\mathbf{z} = \mathbf{Vx}$ with $\mathbf{V}$ the whitening matrix. Assume that the dimensionality of $\mathbf{z}$ has been reduced to that of $\mathbf{s}$ in the whitening, which is always possible in the overdetermined case (number of sensors is not smaller than number of sources).

It holds now $\mathbf{z} = \mathbf{VAs}$. Because both $\mathbf{z}$ and $\mathbf{s}$ have unit covariance matrices (for $\mathbf{s}$, this is assumed in Theorem 1), the matrix $\mathbf{VA}$ must be square orthogonal. This holds even in the case when $\mathbf{s}$ and $\mathbf{z}$ have non-zero means. We can write

$$\mathbf{y} = \mathbf{Qs} = \mathbf{Q}(\mathbf{VA})^T\mathbf{z} = \mathbf{Wz}$$

where the matrix $\mathbf{W}$ is a new parametrization of the problem. The key fact is that $\mathbf{W}$ is orthogonal, because it is the product of two orthogonal matrices $\mathbf{Q}$ and $(\mathbf{VA})^T$. By Theorem 1, to find the sources, it now suffices to *find an orthogonal matrix* $\mathbf{W}$ *for which* $\mathbf{y} = \mathbf{Wz}$ *is non-negative.* The elements of $\mathbf{y}$ are then the sources.

It was further suggested by [10] that a suitable cost function for actually finding the rotation could be constructed as follows: suppose we have an output truncated at zero, $\mathbf{y}^+ = (y_1^+, ..., y_n^+)$ with $y_i^+ = \max(0, y_i)$, and we construct a reestimate of $\mathbf{z} = \mathbf{W}^T\mathbf{y}$ given by $\hat{\mathbf{z}} = \mathbf{W}^T\mathbf{y}^+$. Then a suitable cost function would be given by

$$J(\mathbf{W}) = E\{\|\mathbf{z} - \hat{\mathbf{z}}\|^2\} = E\{\|\mathbf{z} - \mathbf{W}^T\mathbf{y}^+\|^2\}. \quad (1)$$

Due to the orthogonality of matrix $\mathbf{W}$, this is in fact equal to

$$J(\mathbf{W}) = E\{\|\mathbf{y} - \mathbf{y}^+\|^2\} = \sum_{i=1}^{n} E\{\min(0, y_i)^2\}. \quad (2)$$

Obviously, the value will be zero if $\mathbf{W}$ is such that all the y_i are positive.

The minimization of this cost function by various numerical algorithms was suggested in [11, 12, 7]. In [11], explicit axis rotations as well as geodesic search over the Stiefel manifold of orthogonal matrices were used. In [12], the cost function (1) was taken as a special case of "nonlinear PCA" for which an algorithm was earlier suggested by one of the authors [6]. Finally, in [7], it was shown that the cost function (2) is a Liapunov funtion for a certain matrix flow in the Stiefel manifold, providing global convergence.

However, the problem with the gradient type of learning rules is slow speed of convergence. It would be tempting therefore to develop a "fast" numerical algorithm for this problem, perhaps along the lines of the well-known FastICA method [3]. In this paper, such an algorithm is suggested and its convergence is theoretically analyzed.

2 The Classical FastICA Algorithm

Under the whitened zero-mean demixing model $\mathbf{y} = \mathbf{Wz}$, the classical FastICA algorithm finds the extrema of a generic cost function $E\{G(\mathbf{w}^T\mathbf{z})\}$, where $\mathbf{w}^T$

is one of the rows of the demixing matrix $\mathbf{W}$. The cost function can be e.g. a normalized cumulant or an approximation of the marginal entropy which is minimized in ICA in order to find maximally nongaussian projections $\mathbf{w}^T\mathbf{z}$. Under fairly weak assumptions, the true independent sources are among the extrema of $E\{G(\mathbf{w}^T\mathbf{z})\}$ [3]. FastICA updates $\mathbf{w}$ according to the following rule:

$$\mathbf{w} \leftarrow E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} - E\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w}. \tag{3}$$

Here g is the derivative of G, and g' is the derivative of g. After (3), the vectors $\mathbf{w}$ are orthogonalized either in a deflation mode or symmetrically. The algorithm typically converges in a small number of steps to a demixing matrix $\mathbf{W}$, and $\mathbf{y}$ becomes a permutation of the source vector $\mathbf{s}$ with arbitrary signs.

3 The Non-negative FastICA Algorithm

For the non-negative independent components, our task becomes to find an orthogonal matrix $\mathbf{W}$ such that $\mathbf{y} = \mathbf{W}\mathbf{z}$ is nonnegative with the pre-whitened vector $\mathbf{z}$.

The classical FastICA is now facing two problems. First, the non-negative sources cannot have zero means. The mean values must be explicitly included in the analysis. Second, in FastICA, the function g in equation (3) is assumed to be an odd fuction, the derivative of the even function G. If this condition fails to be satisfied, the FastICA as such may not work. Applying FastICA to minimizing the cost function (2), we see that $G(y) = min(0, y)^2$ whose negative derivative (dropping the 2) is

$$g_-(y) = -min(0, y) = \begin{cases} -y, & y < 0 \\ 0, & y \geq 0. \end{cases} \tag{4}$$

We see that it does not satisfy the condition for FastICA.

In order to correct these problems, first, we use non-centered but whitened data $\mathbf{z}$, which satisfies $E\{(\mathbf{z}-E\{\mathbf{z}\})(\mathbf{z}-E\{\mathbf{z}\})^T\} = \mathbf{I}$. Second, we add a control parameter μ on the FastICA update rule (3), giving the following update rule:

$$\mathbf{w} \leftarrow E\{(\mathbf{z} - E\{\mathbf{z}\})g_-(\mathbf{w}^T\mathbf{z})\} - \mu E\{g'_-(\mathbf{w}^T\mathbf{z})\}\mathbf{w}, \tag{5}$$

where g'_- is the derivative of g_-. This formulation shows the similarity to the classical FastICA algorithm. Substituting function g_- from (4) simplifies the terms; for example, $E\{g'_-(\mathbf{w}^T\mathbf{z})\} = -E\{1|\mathbf{w}^T\mathbf{z} < 0\}P\{\mathbf{w}^T\mathbf{z} < 0\}$. The scalar $P\{\mathbf{w}^T\mathbf{z} < 0\}$, appearing in both terms in (5), can be dropped because the vector $\mathbf{w}$ will be normalized anyway. In practice, expectations are replaced by sample averages.

In (5), μ is a parameter determined by:

$$\mu = \min_{\{\mathbf{z}:\mathbf{z}\in\Delta)\}} \frac{E\{(\mathbf{z} - E\{\mathbf{z}\})\mathbf{w}^T\mathbf{z}|\mathbf{w}^T\mathbf{z} < 0\}^T\mathbf{z}}{E\{1|\mathbf{w}^T\mathbf{z} < 0\}\mathbf{w}^T\mathbf{z}}. \tag{6}$$

There the set $\Delta = \{\mathbf{z} : \mathbf{z}^T\mathbf{z}(0) = 0\}$, with $\mathbf{z}(0)$ the vector satisfying $||\mathbf{z}(0)|| = 1$ and $\mathbf{w}^T\mathbf{z}(0) = \max(\mathbf{w}^T\mathbf{z})$. Computing this parameter is computationally somewhat heavy, but on the other hand, now the algorithm converges in a fixed number of steps.

The nonnegative FastICA algorithm is shown in Table 1.

Table 1. The Non-negative FastICA algorithm for estimating several ICs.

1. Whiten the data to get vector $\mathbf{z}$.
2. Set counter $p \leftarrow 1$.
3. Choose an initial vector $\mathbf{w}_p$ of unit norm, and orthogonalize it as

$$\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{j=1}^{p-1} (\mathbf{w}_p^T \mathbf{w}_j)\mathbf{w}_j$$

   and then normalize by $\mathbf{w}_p \leftarrow \mathbf{w}_p / ||\mathbf{w}_p||$.
4. If $\max_{\mathbf{z} \neq 0}(\mathbf{w}_p^T\mathbf{z}) \leq 0$, update $\mathbf{w}_p$ by $-\mathbf{w}_p^T$.
5. If $\min_{\mathbf{z} \neq 0}(\mathbf{w}_p^T\mathbf{z}) \geq 0$, update $\mathbf{w}_p$ by the vector orthogonal to $\mathbf{w}_p$ and the source vectors that are orthogonal to $\mathbf{w}_p$. (See equation (11)).
6. Update $\mathbf{w}_p$ by the equation (5), replacing expectations by sample averages.
7. Let $\mathbf{w}_p \leftarrow \mathbf{w}_p / ||\mathbf{w}_p||$.
8. If $\mathbf{w}_p$ has not converged, go back to step (4).
9. Set $p \leftarrow p + 1$. If $p < n$ where n is the number of independent components, go back to step (3).

4 Analysis of the Algorithm

To make use of the properties of the non-negative independent sources, we perform the following orthogonal variable change:

$$\mathbf{q} = \mathbf{A}^T\mathbf{V}^T\mathbf{w} \tag{7}$$

where $\mathbf{A}$ is the mixture matrix and $\mathbf{V}$ is the whitening matrix. Then

$$\mathbf{w}^T\mathbf{z} = \mathbf{q}^T(\mathbf{VA})^T(\mathbf{VAs}) = \mathbf{q}^T\mathbf{s}. \tag{8}$$

Remember that matrix $\mathbf{VA}$ is orthogonal.

Our goal is to find the orthogonal matrix $\mathbf{W}$ such that $\mathbf{Wz}$ is non-negative. This is equivalent to finding a permutation matrix $\mathbf{Q}$, whose rows will be denoted by vectors $\mathbf{q}^T$, such that $\mathbf{Qs}$ is non-negative. In the space of the $\mathbf{q}$ vectors, the convergence result of the non-negative FastICA algorithm must be a unit vector $\mathbf{q}$ with exactly one entry nonzero and equal to one.

4.1 The Proof of the Convergence

Using the above transformation of eq. (7), the definition of the function g_-, and the parameter μ, the update rule (5) for the variable $\mathbf{q}$ becomes

$$\mathbf{q} \leftarrow \mu E\{1|\mathbf{q}^T\mathbf{s} < 0\}\mathbf{q} - E\{(\mathbf{s} - E\{\mathbf{s}\})(\mathbf{q}^T\mathbf{s})|\mathbf{q}^T\mathbf{s} < 0\}. \tag{9}$$

Before each iteration, there are three cases for $\mathbf{q}^T\mathbf{s}$. If $\mathbf{q}^T\mathbf{s} \leq 0$ for all the sources $\mathbf{s}$, that is, $\mathbf{q} \leq 0$, we simply update it by $\mathbf{q} \leftarrow -\mathbf{q}$ as shown in step 4 in the algorithm. So we only need to consider the other two cases, $\mathbf{q} \geq 0$, and $\min(\mathbf{q}^T\mathbf{s}) < 0$ with $\max(\mathbf{q}^T\mathbf{s}) > 0$.

A. Consider the case that $\min(\mathbf{q}^T\mathbf{s}) < 0$ *and* $\max(\mathbf{q}^T\mathbf{s}) > 0$. Since the sources are positive, then at least one element of $\mathbf{q}$ is negative, and one element is positive. Let $\mathbf{q}(k)$ be the vector after kth iteration. It is easy to see that the update equation (9) keeps zero elements unvariable, that is, $\mathbf{q}(k+1)_i = 0$ if $\mathbf{q}(k)_i = 0$. This can be shown from equation (9)

$$\begin{aligned}\mathbf{q}(k+1)_i &= \mu E\{\mathbf{q}(k)_i|\mathbf{q}(k)^T\mathbf{s} < 0\} - E\{(\mathbf{s}_i - E\{\mathbf{s}_i\})(\mathbf{q}(k)^T\mathbf{s})|\mathbf{q}(k)^T\mathbf{s} < 0\} \\ &= 0 - E\{\mathbf{s}_i - E\{\mathbf{s}_i\}\}E\{(\mathbf{q}(k)^T\mathbf{s})|\mathbf{q}(k)^T\mathbf{s} < 0\} = 0\end{aligned}$$

by noticing that $\mathbf{s}_i$ is independent to $\mathbf{q}(k)^T\mathbf{s} = \sum_{j\neq i}\mathbf{q}(k)_j\mathbf{s}_j$.

Let I and J be the index sets such that $\mathbf{q}(k)_i < 0$ for all $i \in I$ and $\mathbf{q}(k)_j > 0$ for all $j \in J$. Let $\mathbf{s}(0)$ be the source vector such that $\mathbf{q}(k)^T\mathbf{s}(0) = \max(\mathbf{q}(k)^T\mathbf{s})$ and $||\mathbf{s}(0)|| = 1$. The vector $\mathbf{s}(0)$ exists and $\mathbf{s}(0)_i = 0$ for $i \in I$. Further, let the source set $\Delta' := \{\mathbf{s} : \mathbf{s}^T\mathbf{s}(0) = 0\}$, which is not empty; we have for all $\mathbf{s} \in \Delta'$, $\mathbf{s}_j = 0$ for $j \in J$.

By the equation (6) and the transformation equation (7), we have the parameter estimation with variable $\mathbf{q}$

$$\mu = \min_{\{\mathbf{s}:\mathbf{s}\in\Delta'\}} \frac{E\{(\mathbf{s} - E\{\mathbf{s}\})\mathbf{q}(k)^T\mathbf{s}|\mathbf{q}(k)^T\mathbf{s} < 0\}^T\mathbf{s}}{E\{1|\mathbf{q}(k)^T\mathbf{s} < 0\}\mathbf{q}(k)^T\mathbf{s}}. \tag{10}$$

Then for $\mathbf{s} \in \Delta'$, $\mathbf{q}^T\mathbf{s} < 0$ and

$$\mu \leq \frac{E\{(\mathbf{s} - E\{\mathbf{s}\})\mathbf{q}(k)^T\mathbf{s}|\mathbf{q}(k)^T\mathbf{s} < 0\}^T\mathbf{s}}{E\{1|\mathbf{q}(k)^T\mathbf{s} < 0\}\mathbf{q}(k)^T\mathbf{s}}.$$

Therefore, $\mathbf{q}(k+1)^T\mathbf{s} \geq 0$.

Since $\mathbf{e}_i s_i$ belongs to the set Δ' if $i \in I$, where $\mathbf{e}_i$ is the unit vector with the ith entry one and the others zero, it holds $\mathbf{q}(k+1)^T\mathbf{e}_i s_i \geq 0$. This implies that $\mathbf{q}(k+1)_i \geq 0$ for $i \in I$. According to the choice of parameter μ, there exists at least one source $\mathbf{s} \in \Delta'$ such that $\mathbf{q}(k+1)^T\mathbf{s} = 0$, that is $\sum_{\{i\in I\}}\mathbf{q}(k+1)_i s_i = 0$. Since the sources are nonnegative, and also for $i \in I$, $\mathbf{q}(k+1)_i$ is nonnegative, there is at least one index $i_0 \in I$, such that $\mathbf{q}(k+1)_{i_o} = 0$. Therefore, after this iteration, the number of zero elements of vector $\mathbf{q}$ increases.

B. Consider the case that $\mathbf{q} \geq 0$. Then, the algorithm updates $\mathbf{q}$ by the orthogonal vector of $\mathbf{q}$ which keeps the zero elements of $\mathbf{q}$ zero. Since $\mathbf{q} \geq 0$, its orthogonal vector will not be nonnegative or negative, and the iteration goes back to the case A we just discussed.

To find this update vector, consider the sources $\hat{\mathbf{S}} := \{\mathbf{s} \neq 0 : \mathbf{q}(k)^T\mathbf{s} = 0\}$. The updated vector $\mathbf{q}(k+1)$ can be chosen as the vector, which is orthogonal to $\mathbf{q}(k)$ and $\hat{\mathbf{S}}$. To do this, let all the vectors in the sources space $\hat{\mathbf{S}}$ be column vectors forming matrix $\mathbf{B}$. Then the null space $\mathrm{null}(\mathbf{B})$ is orthogonal to the sources space $\hat{\mathbf{S}}$. If null($\mathbf{B}$) contains only one column, then this column vector is what we want, and the iteration goes to next step. Otherwise, take any column $\mathbf{q}(r)$ from null($\mathbf{B}$) which is different from $\mathbf{q}(k)$, and the update rule is

$$\mathbf{q}(k+1) = \mathbf{q}(r) - (\mathbf{q}(r)^T \mathbf{q}(k))\mathbf{q}(k). \tag{11}$$

Therefore, after each iteration, the updated vector $\mathbf{q}$ keeps the old zero entries zero and gains one more zero entry. So within $n-1$ iteration steps, the vector $\mathbf{q}$ is updated to be a unit vector $\mathbf{e}_i$ for certain i. With total iterative steps $\sum_{i=1}^{n-1} i = n(n-1)/2$, the permutation matrix $\mathbf{Q}$ is formed.

4.2 Complexity of the Computation

As the analysis in the above section shows, the total iteration steps of our algorithm are less than or equal to $n(n-1)/2$. During each iteration, the computational differences compared to classic FastICA come from step 4, 5 and 6 as shown in Table 1. The step 4 does not increase the computation much, so we can almost omit it. In step 5, we need to calculate the value of $\mathbf{w}_p^T \mathbf{z}$ once, and solve a $m \times n$ line equation (m is the number of vectors in the source space $\{\mathbf{s} \neq 0 : \mathbf{q}(k)^T \mathbf{s} = 0\}$). This can be solved by Matlab command null() immediately. Step 6 is the main update rule, just as in FastICA, and we need to calculate the expectation $E\{\mathbf{z} - E\{\mathbf{z}\}\}$. Furthermore, in our algorithm, to calculate the parameter μ, we need to go through the data $\mathbf{z}$ once more.

5 Experiments

In this section we present some numerical simulations, run in Matlab, to demonstrate the behaviour of the algorithm. The demixing matrix $\mathbf{W}$ is initialized to the identity matrix, ensuring initial orthogonality of $\mathbf{W}$ and hence of $\mathbf{H} = \mathbf{WVA}$.

The algorithm was applied to an image unmixing task. 4 image patches of size 252×252 were selected from a set of images of natural scenes, and downsampled by a factor of 4 in both directions to yield 63×63 images (see [7]). Each of the $n = 4$ images was treated as one source, with its pixel values representing the $63 \times 63 = 3969$ samples. The source image values were shifted to have a minimum of zero, to ensure they were *well-grounded* as required by the algorithm, and the images were scaled to ensure they were all unit variance.

After scaling, the source covariance matrix $\overline{\mathbf{s}\mathbf{s}^T} - \bar{\mathbf{s}}\bar{\mathbf{s}}^T$ was computed and the largest off-diagonal element turned out to be 0.16. This is an acceptably small covariance between the images: as with any ICA method based on prewhitening, any covariance between sources would prevent accurate identification of the sources. A mixing matrix $\mathbf{A}$ was generated randomly and used to construct the four mixture images.

After iteration, the source-to-output matrix $\mathbf{H} = \mathbf{WVA}$ was

$$\mathbf{H} = \begin{pmatrix} 0.058 & \mathbf{1.010} & -0.106 & 0.062 \\ -0.106 & 0.042 & -0.078 & \mathbf{1.002} \\ -0.003 & -0.017 & \mathbf{1.014} & 0.076 \\ \mathbf{0.997} & -0.105 & -0.102 & -0.086 \end{pmatrix} \tag{12}$$

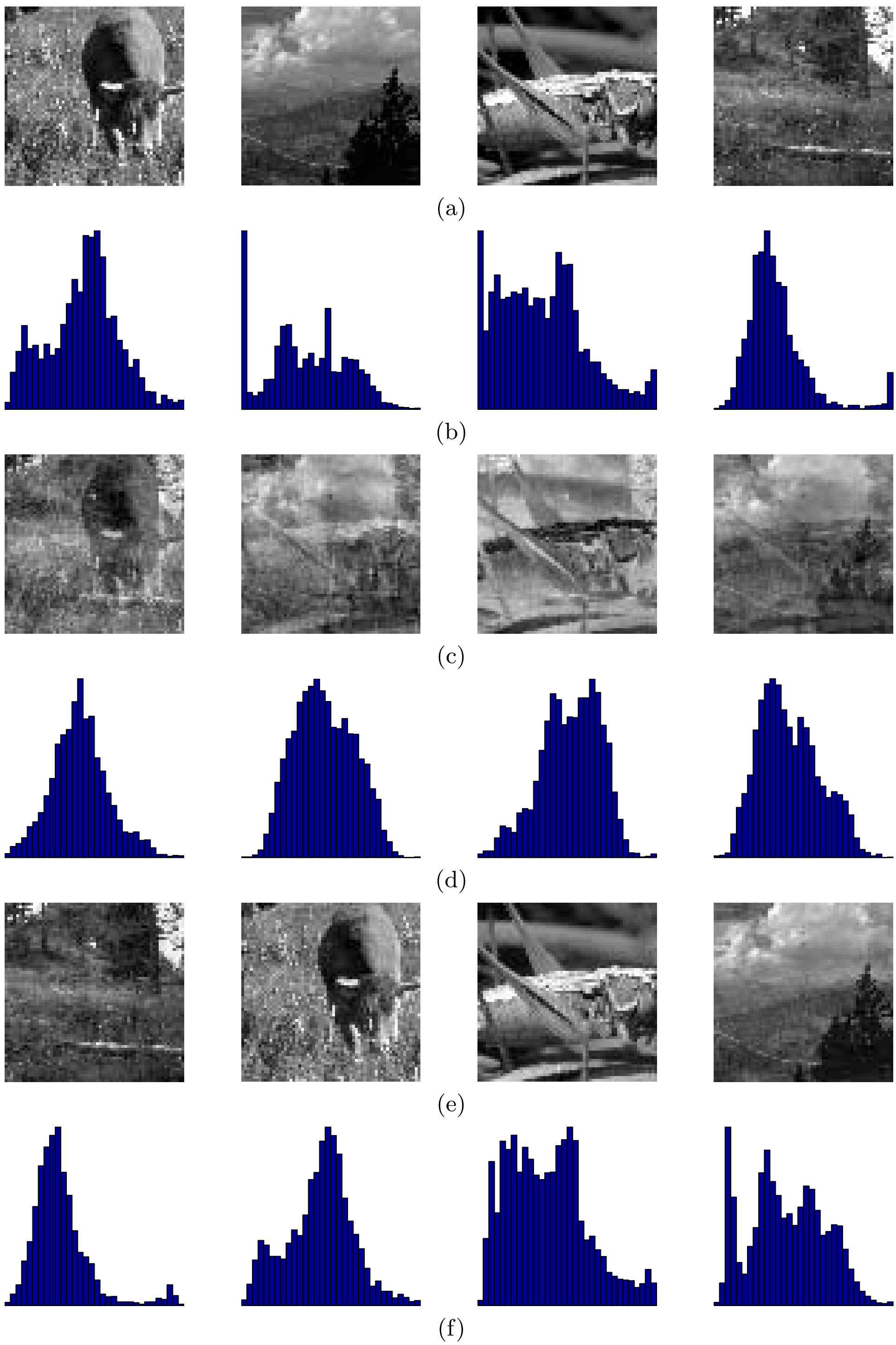

Fig. 1. Images and histograms for the image separation using the non-negative FastICA algorithm, showing (a) source images and (b) their histograms, (c), (d) the mixed images and their histograms, and (e), (f) the separated images and their histograms.

Figure 1 shows the original, mixed and separated images and their histograms. The algorithm converges in 6 steps and is able to separate the images reasonably well.

References

1. S-I. Amari and A. Cichocki. *Adaptive Blind Signal and Image Processing.* John Wiley & sons, Inc., 2002.
2. Ronald C. Henry. Multivariate receptor models–current practice and future trends. *Chemometrics and Intelligent Laboratory Systems*, 60(1-2):43–48, January 2002.
3. A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley & Sons, 2001.
4. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 21 October 1999.
5. Jae Sung Lee, Daniel D. Lee, Seugjin Choi, and Dong Soo Lee. Application of non-negative matrix factorization to dynamic positron emission tomography. In T.-W. Lee, T.-P. Jung, S. Makeig, and T. J. Sejnowski, editors, *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA2001), San Diego, California*, pages 629–632, December 9-13 2001.
6. E. Oja. The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1):25–46, 1997.
7. E. Oja and M. D. Plumbley. Blind separation of positive sources by globally convergent gradient search. *Neural Computation*, 16, 2004.
8. P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.
9. Lucas Parra, Clay Spence, Paul Sajda, Andreas Ziehe, and Klaus-Robert Müller. Unmixing hyperspectral data. In *Advances in Neural Information Processing Systems 12 (Proc. NIPS*99)*, pages 942–948. MIT Press, 2000.
10. M. D. Plumbley. Conditions for nonnegative independent component analysis. *IEEE Signal Processing Letters*, 9(6):177 –180, June 2002.
11. M. D. Plumbley. Algorithms for non-negative independent component analysis. *IEEE Transaction on Neural Networks*, 14(3):534 –543, 2003.
12. M. D. Plumbley and E. Oja. A 'non-negative PCA' algorithm for independent component analysis. *IEEE Transactions on Neural Networks*, 15(1):66 – 76, 2004.
13. S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita. Dimensionality reduction using non-negative matrix factorization for information retrieval. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 960 – 965 vol.2, Tucson, AZ, USA, 7-10 October 2001.