# Alternative derivation of FastICA with novel power iteration algorithm

Shahab Basiri, Esa Ollila, *Member, IEEE,* and Visa Koivunen, *Fellow, IEEE*

***Abstract*—The widely used fixed-point FastICA algorithm has been derived and motivated as being an *approximate* Newton-Raphson (NR) algorithm. In the original derivation, the Lagrangian multiplier is treated as a constant and an ad-hoc approximation is used for Jacobian matrix in the NR update. In this paper, we provide an alternative derivation of the FastICA algorithm without approximation. We show that any solution to the FastICA algorithm is a solution to the exact NR algorithm as well. In addition, we propose a novel *power iteration* algorithm for FastICA which is remarkably more stable than the fixed-point algorithm, when the sample size is not orders of magnitudes larger than the dimension. Our proposed algorithm can be run on parallel computing nodes.**

***Index Terms*—Independent Component Analysis; FastICA; Newton-Raphson method; Power method**

## I. Introduction

Independent component analysis (ICA) [1]–[3] is a widely used signal processing technique in extracting unobserved independent source signals from their observed multivariate mixture recordings. The FastICA fixed-point algorithm [1], [2] is one of the most popular ICA algorithms.

The derivation of the FastICA algorithm [1] requires that the data is centered and pre-whitened, so that we have equal number of mixtures as there are unknown sources. The observed whitened random vector $\mathbf{x} \in \mathbb{R}^d$ is then a *linear* mixture of the unobserved random *source* vector $\mathbf{s} = (s_1, \ldots, s_d)^\top$ possessing statistically independent components (IC's), i.e.,

$$\mathbf{x} = \mathbf{W}^\top \mathbf{s} = \mathbf{w}_1 s_1 + \cdots + \mathbf{w}_d s_d, \tag{1}$$

where the unknown $d \times d$ mixing matrix $\mathbf{W}^\top = \begin{pmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_d \end{pmatrix}$ is an orthogonal matrix. Due to whitening and centering, we have that $\mathbb{E}[\mathbf{s}] = \mathbf{0}$ and $\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbf{I}$. Note that since there is a scale ambiguity in solving the ICA model, it is assumed without loss of generality that $\mathbb{E}[s_i^2] = 1$, $i = 1, \ldots, d$. Furthermore, the mixing matrix of the whitened data is orthogonal i.e. $\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbf{W}^\top \mathbb{E}[\mathbf{s}\mathbf{s}^\top]\mathbf{W} = \mathbf{W}^\top \mathbf{W} = \mathbf{I}$.

The *1-unit* FastICA estimator finds a demixing vector $\mathbf{w}$ as a local maxima of a non-Gaussianity measure $\left|\mathbb{E}\big[G(\mathbf{w}^\top \mathbf{x})\big]\right|$ under the unit-norm constraint $\left\|\mathbf{w}\right\|^2 = \mathbf{w}^\top \mathbf{w} = 1$, where $G$ can be any twice continuously differentiable nonlinear and non-quadratic function with $G(0) = 0$. Thus the *1-unit* FastICA estimator maximizes the Lagrangian

$$\mathcal{L}(\mathbf{w}; \lambda) = \left|\mathbb{E}\big[G(\mathbf{w}^\top \mathbf{x})\big]\right| - \frac{\lambda}{2}(\mathbf{w}^\top \mathbf{w} - 1), \tag{2}$$

where $\lambda$ is the Lagrange multiplier. We write $g = G'$ and $g' = G''$ for the 1st and 2nd derivative of $G$ respectively, where $g$ is referred to as ICA *nonlinearity*. The local optimum of (2) verifies the following estimating equation, which is obtained by setting the gradient of the Lagrangian w.r.t. $\mathbf{w}$ to zero.

$$F(\mathbf{w}) = \mathbf{m}(\mathbf{w}) - \lambda(\mathbf{w})\mathbf{w} = \mathbf{0}, \tag{3}$$

where $\mathbf{m}(\mathbf{w}) = \mathbb{E}[g(\mathbf{w}^\top \mathbf{x})\mathbf{x}]$ and $\lambda(\mathbf{w}) = \mathbf{w}^\top \mathbf{m}$ is obtained by multiplying both sides of (3) by $\mathbf{w}^\top$ from the left. The 1-unit fixed-point FastICA algorithm in [1] is motivated as being an approximate NR update for solving (3). The algorithm iterates

$$\mathbf{w} \leftarrow \frac{\mathbf{m}(\mathbf{w}) - \beta(\mathbf{w})\mathbf{w}}{\|\mathbf{m}(\mathbf{w}) - \beta(\mathbf{w})\mathbf{w}\|} \tag{4}$$

until convergence. The term $\beta(\mathbf{w})$ in (4) is a scalar multiplier defined as $\beta(\mathbf{w}) = \mathbb{E}[g'(\mathbf{w}^\top \mathbf{x})] \in \mathbb{R}$.

In the original derivation of the FastICA algorithm [1] and [2, C. 8 p. 189] the Lagrangian and the Jacobian matrix are oversimplified using unnecessary assumptions. To be more specific, in the NR formulation, the Lagrangian multiplier $\lambda(\mathbf{w})$ is treated as a constant that does not depend on $\mathbf{w}$. Due to this simplification, a post-normalization step is necessary in order to keep the NR update in the feasible set. Also, an ad-hoc approximation of the Jacobian matrix of $F(\mathbf{w})$ is utilized that does not necessarily have a clear statistical justification.

In this paper, we provide an alternative derivation of the fixed-point FastICA algorithm which does not require simplifying assumptions. Specifically, we show that any solution to (4) is a solution to the exact NR algorithm as well. Furthermore, our new derivation of the FastICA algorithm leads us to propose a new power iteration (PI) method for FastICA which is shown to be significantly more stable than the original FastICA algorithm. The proposed PI method always converges to a valid solution even if the common case of $d \ll n$ is not valid, i.e. the dimensionality and the number of observations are of the same order. This is the finite-sample regime in which the FastICA algorithm is often reported to have convergence problems; See Table I in [4], Table III - VI in [5] and Table I in this paper. Our proposed method can be run on parallel computing nodes, which drastically reduces the computational time. This may not be possible with the FastICA algorithm.

The paper is organized as follows. In Section II, the original derivation of the FastICA algorithm in (4) is reviewed and the underlying assumptions are discussed. In Section III, a novel derivation of the FastICA algorithm is provided. In Section IV we view the FastICA algorithm as a power iteration (PI) method. This leads to new power iteration FastICA algorithm which is described in Section V. Section VI provides numerical examples and Section VII concludes the paper.

## II. Original derivation of the FastICA algorithm

A compact overview of the original derivation of FastICA algorithm [1], [2] is provided. Furthermore, we point out some oversimplifying assumptions that can be relaxed.

The Newton-Rapshon update for solving $F(\mathbf{w}) = \mathbf{0}$ is

$$\mathbf{w} \leftarrow \mathbf{w} - [J_F(\mathbf{w})]^{-1} F(\mathbf{w}), \tag{5}$$

where $J_F$ denotes the Jacobian of $F(\cdot)$ in (3) w.r.t. $\mathbf{w}$.

First, the Lagrangian multiplier $\lambda(\mathbf{w}) = \mathbf{w}^\top \mathbf{m}$ in (3) is treated as a constant that does not depend on $\mathbf{w}$. As a consequence, an additional post-normalization step is necessary to keep the update in the feasible set:

$$\begin{cases} \mathbf{w} \leftarrow \mathbf{w} - [\mathbf{M}(\mathbf{w}) - \lambda \mathbf{I}]^{-1} F(\mathbf{w}) \\ \mathbf{w} \leftarrow \mathbf{w} / \left\| \mathbf{w} \right\| \end{cases}, \tag{6}$$

where $\mathbf{M}(\mathbf{w}) = \mathbb{E}[g'(\mathbf{w}^\top \mathbf{x}) \mathbf{x}\mathbf{x}^\top]$ and $[\mathbf{M}(\mathbf{w}) - \lambda \mathbf{I}]$ is the Jacobian of $F(\cdot)$ when $\lambda$ is treated as a constant. Furthermore, an ad-hoc approximation for $\mathbf{M}(\mathbf{w})$ is utilized [2, C. 8 p. 189]:

$$\mathbf{M}(\mathbf{w}) \approx \mathbb{E}[g'(\mathbf{w}^\top \mathbf{x})] \mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \beta(\mathbf{w}) \mathbf{I}, \tag{7}$$

where $\beta(\mathbf{w})$ is defined in (4). Note that, this approximation does not have statistical justification. This follows from the property that the expectation operator is not multiplicative unless the involved random variables are independent or at least uncorrelated. By substituting the approximation in (7) to (6), one obtains the following updates:

$$\begin{cases} \mathbf{w} \leftarrow \mathbf{w} - F(\mathbf{w}) / (\beta(\mathbf{w}) - \lambda) \\ \mathbf{w} \leftarrow \mathbf{w} / \left\| \mathbf{w} \right\| \end{cases}, \tag{8}$$

Substituting $F(\mathbf{w})$ from (3) in (8) then results in the 1-unit FastICA algorithm that was given in (4). It has been shown in [1], [2] that the algorithm converges up to a sign ambiguity to one of the rows, say $\mathbf{w}_k$, of the demixing matrix $\mathbf{W}$ given that $\mathbb{E}[s_k g(s_k)] \neq \mathbb{E}[g'(s_k)]$, with $s_k = \mathbf{w}_k^\top \mathbf{x}$. It is not known, however, to which row vector $\mathbf{w}_k$ the algorithm converges to. It usually depends on the initial start of the algorithm.

## III. Alternative derivation

The main contribution of this section is to provide an alternative derivation of the fixed-point FastICA algorithm (4). In our derivation, $\lambda(\mathbf{w}) = \mathbf{w}^\top \mathbf{m}$ is not treated as a constant and the questionable approximation (7) is not used. We show that the solution $\mathbf{w}_k$ of the fixed-point equation (4) is also a solution to the *exact* NR algorithm (5). This fact has not been shown in the literature so far. The following Lemma is needed in our derivations.

**Lemma 1.** *Let* $\mathbf{x}$ *be a random vector following the ICA model* (1) *and define* $\mathbf{M}_k = \mathbf{M}(\mathbf{w}_k) = \mathbb{E}\big[g'\big(s_k\big)\mathbf{x}\mathbf{x}^\top\big] \in \mathbb{R}^{d \times d}$. *Then*

[a] $\mathbf{M}_k = \alpha_k \mathbf{w}_k \mathbf{w}_k^\top + \beta_k \sum_{j=1, j \neq k}^{d} \mathbf{w}_j \mathbf{w}_j^\top$,
*where* $\alpha_k = \mathbb{E}[g'(s_k) s_k^2]$ *and* $\beta_k = \beta(\mathbf{w}_k) = \mathbb{E}\big[g'\big(s_k\big)\big]$.

[b] $(\mathbf{I} - \mathbf{w}_k \mathbf{w}_k^\top)\mathbf{M}_k = \beta_k (\mathbf{I} - \mathbf{w}_k \mathbf{w}_k^\top)$.

*Proof.* [a] Note that $\mathbf{x} = \mathbf{W}^\top \mathbf{s}$, so that

$$\begin{aligned} \mathbb{E}[g'(s_k)\mathbf{x}\mathbf{x}^\top] &= \mathbf{W}^\top \mathbb{E}[g'(s_k)\mathbf{s}\mathbf{s}^\top]\mathbf{W} \\ &= \mathbf{W}^\top \mathrm{diag}(\beta_k, \ldots, \alpha_k, \ldots, \beta_k)\mathbf{W} \\ &= \alpha_k \mathbf{w}_k \mathbf{w}_k^\top + \beta_k \sum_{j \neq k} \mathbf{w}_j \mathbf{w}_j^\top . \end{aligned}$$

Above we used that $\mathbb{E}[g'(s_k) s_i^2] = \mathbb{E}[g'(s_k)]\mathbb{E}[s_i^2] = \mathbb{E}[g'(s_k)]$ for $k \neq i$ and $\mathbb{E}[g'(s_k) s_i s_j] = E[g'(s_k)]\mathbb{E}[s_i s_j] = 0$ as sources are independent and of unit variance. The EVD above also implies that $\mathbf{w}_k^\top \mathbf{M}_k \mathbf{w}_k = \alpha_k$.

[b], The [a]-part of the Lemma 1 yields

$$\begin{aligned} (\mathbf{I} - \mathbf{w}_k \mathbf{w}_k^\top)\mathbf{M}_k &= (\mathbf{I} - \mathbf{w}_k \mathbf{w}_k^\top)\big(\alpha_k \mathbf{w}_k \mathbf{w}_k^\top + \beta_k (\mathbf{I} - \mathbf{w}_k \mathbf{w}_k^\top)\big) \\ &= (\mathbf{I} - \mathbf{w}_k \mathbf{w}_k^\top)\big(\alpha_k \mathbf{w}_k \mathbf{w}_k^\top + \beta_k \mathbf{I}\big) = \beta_k (\mathbf{I} - \mathbf{w}_k \mathbf{w}_k^\top) \end{aligned}$$

which gives the stated result. □

**Corollary 1.** *Using the result of Lemma 1[a], the LHS of* (7) *at the solution* $s_k = \mathbf{w}_k^\top \mathbf{x}$ *is:*

$$\mathbf{M}(\mathbf{w}_k) = \mathbb{E}[g'(s_k)\mathbf{x}\mathbf{x}^\top] = \beta_k \mathbf{I} + (\alpha_k - \beta_k)\mathbf{w}_k \mathbf{w}_k^\top,$$

*where* $(\alpha_k - \beta_k)$ *is the spectral gap between the two eigenvalues of* $\mathbf{M}(\mathbf{w}_k)$. *Such a spectral gap is neglected in the RHS of* (7). *This implies that the approximation* (7) *may not be valid even when* $\mathbf{w}^\top \mathbf{x}$ *is close to* $s_k$.

Since the Lagrange multiplier $\lambda(\mathbf{w}) = \mathbf{w}^\top \mathbf{m}$ depends on the unknown parameter $\mathbf{w}$, the true (non-approximate) Jacobian matrix of $F(\mathbf{w})$ in (3) is

$$J_F(\mathbf{w}) = (\mathbf{I} - \mathbf{w}\mathbf{w}^\top)\mathbf{M}(\mathbf{w}) - \lambda(\mathbf{w})\mathbf{I} - \mathbf{w}\mathbf{m}^\top. \tag{9}$$

For a given solution to (3), $\mathbf{w}_k$, we may use Lemma 1[b] to reformulate (9) as

$$\begin{aligned} &J_F(\mathbf{w}_k) = \beta_k (\mathbf{I} - \mathbf{w}_k \mathbf{w}_k^\top) - \lambda_k \mathbf{I} - \mathbf{w}_k \mathbf{m}_k^\top = \\ &(\beta_k - \lambda_k)\mathbf{I} - \mathbf{w}_k (\beta_k \mathbf{w}_k^\top + \mathbf{m}_k^\top) = (\beta_k - \lambda_k)\mathbf{I} - \mathbf{w}_k \mathbf{v}_k^\top, \end{aligned} \tag{10}$$

where $\mathbf{m}_k = \mathbb{E}[g(\mathbf{w}_k^\top \mathbf{x})\mathbf{x}]$, $\lambda_k = \mathbf{w}_k^\top \mathbf{m}_k$ and $\mathbf{v}_k = (\beta_k \mathbf{w}_k + \mathbf{m}_k)$. Next, we use the Sherman-Morrison matrix inversion lemma [6] to write

$$[J_F(\mathbf{w}_k)]^{-1} = \Big((\beta_k - \lambda_k)\mathbf{I} - \mathbf{w}_k \mathbf{v}_k^\top\Big)^{-1} = \frac{1}{\beta_k - \lambda_k}\Big(\mathbf{I} - \frac{\mathbf{w}_k \mathbf{v}_k^\top}{2\lambda_k}\Big).$$

Thus, for any solution to (3), the term $[J_F(\mathbf{w})]^{-1} F(\mathbf{w})$ in (5) becomes

$$\begin{aligned} [J_F(\mathbf{w}_k)]^{-1} F(\mathbf{w}_k) &= \frac{1}{\beta_k - \lambda_k}\Big(\mathbf{I} - \frac{\mathbf{w}_k \mathbf{v}_k^\top}{2\lambda_k}\Big)(\mathbf{m}_k - \lambda_k \mathbf{w}_k) \\ &= \frac{1}{\beta_k - \lambda_k}(\mathbf{m}_k - \lambda_k \mathbf{w}_k) \ , \end{aligned} \tag{11}$$

where the last identity follows because $\mathbf{v}_k^\top (\mathbf{m}_k - \lambda_k \mathbf{w}_k) = (\beta_k \mathbf{w}_k + \mathbf{m}_k)^\top (\mathbf{m}_k - \lambda_k \mathbf{w}_k) = 0$; This property follows by recalling that $\lambda_k = \mathbf{m}_k^\top \mathbf{w}_k$ and noting that $\mathbf{m}_k^\top \mathbf{m}_k = \lambda_k^2$ which is obtained by multiplying both sides of (3) by $\mathbf{m}_k^\top$. Using (11), the NR update in (5) becomes

$$\mathbf{w}_k - [J_F(\mathbf{w}_k)]^{-1} F(\mathbf{w}_k) = \frac{\mathbf{m}_k - \beta_k \mathbf{w}_k}{\lambda_k - \beta_k}. \tag{12}$$

Note that $\|\mathbf{m}_k - \beta_k \mathbf{w}_k\|^2 = \mathbf{m}_k^\top \mathbf{m}_k + \beta_k^2 \mathbf{w}_k^\top \mathbf{w}_k -$

$2\beta_k \mathbf{m}_k^\top \mathbf{w}_k = (\lambda_k - \beta_k)^2$ where we again used $\mathbf{m}_k^\top \mathbf{m}_k = \lambda_k^2$ and $\lambda_k = \mathbf{m}_k^\top \mathbf{w}_k$. Hence the NR update in (12) becomes

$$\mathbf{w}_k - [J_F(\mathbf{w}_k)]^{-1} F(\mathbf{w}_k) = \pm \frac{\mathbf{m}_k - \beta_k \mathbf{w}_k}{\|\mathbf{m}_k - \beta_k \mathbf{w}_k\|}, \tag{13}$$

which is the solution to the fixed point equation (4). This leads to an observation that any solution to the FastICA fixed point equation (4) is also a solution to the exact NR algorithm (5).

## IV. FastICA as power iteration method

FastICA algorithm can be viewed as a single-vector iteration method such as Power Iteration (PI), the Inverse Iteration (II) and the Rayleigh Quotient Iteration (RQI) [7], [8]. This is not a surprise since many single-vector iteration approaches are known to stem from the Newton-Raphson method [9]. FastICA as PI method is studied in [7]. Below, we provide further intuition on how FastICA works as a PI method.

Let us rewrite (4) as

$$\mathbf{w} \leftarrow \frac{[\mathbf{H}(\mathbf{w}) - \beta(\mathbf{w})\mathbf{I}]\mathbf{w}}{\|[\mathbf{H}(\mathbf{w}) - \beta(\mathbf{w})\mathbf{I}]\mathbf{w}\|}, \tag{14}$$

where $\mathbf{H}(\mathbf{w}) = \mathbb{E}\big[\frac{g(\mathbf{w}^\top \mathbf{x})}{\mathbf{w}^\top \mathbf{x}} \mathbf{x}\mathbf{x}^\top\big] \in \mathbb{R}^{d \times d}$ is positive definite for all conventional ICA nonlinearities including $pow3$, $tanh$ and $gauss$ [7]. FastICA algorithm in (14) resembles a PI method that starts with an initial guess and finds $\mathbf{w}_k$ as an eigenvector of matrix $[\mathbf{H}(\mathbf{w}_k) - \beta(\mathbf{w}_k)\mathbf{I}]$. Such an eigenvector is a local maximizer of its corresponding eigenvalue in magnitude [7]. Note that $[\mathbf{H}(\mathbf{w}) - \beta(\mathbf{w})\mathbf{I}]$ and $\mathbf{H}(\mathbf{w})$ share the same eigenvectors. Also, the eigenvalues of $[\mathbf{H}(\mathbf{w}) - \beta(\mathbf{w})\mathbf{I}]$ are identical to eigenvalues of $\mathbf{H}(\mathbf{w})$ shifted by a scalar $\beta(\mathbf{w})$. In the following lemma, we show that $\mathbf{H}(\mathbf{w})$ (resp. $[\mathbf{H}(\mathbf{w}) - \beta(\mathbf{w})\mathbf{I}]$) possesses an eigenvector close to $\mathbf{w}$. In addition, the eigenvalue corresponding to $\mathbf{w}$, is farther apart from the bulk of other eigenvalues.

**Lemma 2.** *Let $\mathbf{x}$ be a random vector following the ICA model (1) and let $\tilde{s} = \tilde{\mathbf{w}}^\top \mathbf{x}$, where $\tilde{\mathbf{w}} \in \mathbb{R}^d$ is an arbitrary unit vector. Then $\mathbf{H}(\tilde{\mathbf{w}}) = \mathbb{E}\big[(g(\tilde{s})/\tilde{s})\mathbf{x}\mathbf{x}^\top\big]$ can be approximated by the following EVD.*

$$\mathbf{H}(\tilde{\mathbf{w}}) \approx \mathbb{E}[g(\tilde{s})\tilde{s}]\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top + \mathbb{E}[g(\tilde{s})/\tilde{s}](\mathbf{I} - \tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top).$$

*Proof.* Let $\tilde{\mathbf{W}}^\top = (\tilde{\mathbf{w}}_1 \cdots \tilde{\mathbf{w}}_k \cdots \tilde{\mathbf{w}}_d)$ be an arbitrary orthonormal matrix and define $\tilde{\mathbf{s}} = \tilde{\mathbf{W}}\mathbf{x}$. Note that, $\mathbb{E}[\tilde{\mathbf{s}}] = \mathbf{0}$. Since $\mathbf{x}$ is pre-whitened, $\tilde{\mathbf{s}}$ possesses statistically uncorrelated components $(\tilde{s}_1 \cdots \tilde{s}_k \cdots \tilde{s}_d)^\top$ i.e., $\mathbb{E}[\tilde{\mathbf{s}}\tilde{\mathbf{s}}^\top] = \mathbf{I}$. We set $f(\tilde{s}_k) = g(\tilde{s}_k)/\tilde{s}_k$ and proceed as follows.

$$\mathbf{H}(\tilde{\mathbf{w}}_k) = \tilde{\mathbf{W}}^\top \mathbb{E}[f(\tilde{s}_k)\tilde{\mathbf{s}}\tilde{\mathbf{s}}^\top]\tilde{\mathbf{W}} = \mathbb{E}[g(\tilde{s}_k)\tilde{s}_k]\tilde{\mathbf{w}}_k\tilde{\mathbf{w}}_k^\top + \sum_{j \neq k} \mathbb{E}[f(\tilde{s}_k)\tilde{s}_j^2]\tilde{\mathbf{w}}_j\tilde{\mathbf{w}}_j^\top + \sum_j \sum_{i \neq j} \mathbb{E}[f(\tilde{s}_k)\tilde{s}_j\tilde{s}_i]\tilde{\mathbf{w}}_j\tilde{\mathbf{w}}_i^\top. \tag{15}$$

Using the following approximations in (15) concludes the proof:

1) $\mathbb{E}[f(\tilde{s}_k)\tilde{s}_j^2] \approx \mathbb{E}[f(\tilde{s}_k)]\mathbb{E}[\tilde{s}_j^2] = \mathbb{E}[f(\tilde{s}_k)]$ for $j \neq k$,
2) $\mathbb{E}[f(\tilde{s}_k)\tilde{s}_j\tilde{s}_i] \approx \mathbb{E}[f(\tilde{s}_k)]\mathbb{E}[\tilde{s}_j\tilde{s}_i] = 0$.

Below, we show that the above approximations are sensible. The LHS of the first approximations can be written as :

$$\mathbb{E}[(f(\tilde{s}_k)\tilde{s}_j^2] = \mathbb{E}[f(\tilde{s}_k)]\mathbb{E}[\tilde{s}_j^2] + \mathrm{cov}[f(\tilde{s}_k), \tilde{s}_j^2]. \tag{16}$$

It is sufficient to show that the covariance term in the RHS of (16) is negligible. We use Cauchy-Schwarz inequality to write: $|\mathrm{cov}[f(\tilde{s}_k), \tilde{s}_j^2]| \leq \sqrt{\mathrm{var}[f(\tilde{s}_k)]\mathrm{var}[\tilde{s}_j^2]}$. Using delta method and second-order Taylor expansion (i.e., assuming that the reminder term is negligible), the variance of $f(\tilde{s}_k)$ can be approximated as: $\mathrm{var}[f(\tilde{s}_k)] \approx \Big(f'(\mathbb{E}[\tilde{s}_k])\Big)^2 \mathrm{var}[\tilde{s}_k] = (f'(0))^2$ [10], where $f'(0) = 0$ for all conventional ICA nonlinearities including $pow3$, $tanh$ and $gauss$. Thus, $\mathrm{var}[f(\tilde{s}_k)] \approx 0$ and consequently, $\mathrm{cov}[f(\tilde{s}_k), \tilde{s}_j^2] \approx 0$. The proof of the second approximation is omitted as it is very similar to the first. □

Denote by $\gamma(\mathbf{w})$, the eigenvalue of $\mathbf{H}(\mathbf{w})$ possessing the largest Euclidean distance to the bulk of other eigenvalues. As follows from [7], FastICA algorithm in (14) finds $\mathbf{w}_k$ as a local maximizer of $\delta(\mathbf{w}) = |\gamma(\mathbf{w}) - \beta(\mathbf{w})|$ i.e. $\mathbf{w}_k = \arg\max\{\delta(\mathbf{w})\}$. It is easy to show that, $\delta(\mathbf{w})$ has a global minimum at Gaussian component. More specifically, in the noise-free scenario, suppose that one of the independent components is Gaussian, say $s_k = \mathbf{w}_k^\top \mathbf{x}$. Then, one may use integration by parts, with $u = g(s_k)$ and $v' = s_k e^{-\frac{s_k^2}{2}}$, to show that $\mathbb{E}[g(s_k)s_k] = \mathbb{E}[g'(s_k)]$. This implies that $\delta(\mathbf{w})$ can be viewed as a measure of non-Gaussianity. Later in Section V, we use $\delta(\mathbf{w})$ to assess the superiority of extracted components.

## V. A new stable FastICA algorithm

Here we propose a new power iteration method for FastICA, which is numerically more stable than the original FastICA algorithm.

Recall that $[\mathbf{H}(\mathbf{w}) - \beta(\mathbf{w})\mathbf{I}]$ in (14) and $\mathbf{H}(\mathbf{w})$ share the same eigenvectors. In order to devise an algorithm insensitive to finite sample errors, we do not utilized the spectral shift in (14). Instead, we devise two parallel PI methods that start with the same initial guess and find $\mathbf{w}_{k1}$ and $\mathbf{w}_{k2}$ as a local maximizer and a minimizer of $\gamma(\mathbf{w})$ respectively. Then we assess the superiority of the two extracted components using a measure of non-Gaussianity and discard the one that is closer to Gaussianity.

For all conventional ICA nonlinearities including $pow3$, $tanh$ and $gauss$, $\mathbf{H}(\mathbf{w})$ is positive definite, i.e. $\gamma(\mathbf{w}) > 0$ [7]. Hence, a power iteration of the form

$$\mathbf{w} \leftarrow \frac{\mathbf{H}(\mathbf{w})\mathbf{w}}{\|\mathbf{H}(\mathbf{w})\mathbf{w}\|} = \frac{\mathbf{m}(\mathbf{w})}{\|\mathbf{m}(\mathbf{w})\|}, \tag{17}$$

finds $\mathbf{w}_{k1}$ as a local maximizer of $\gamma(\mathbf{w})$, where $\mathbf{m}$ is defined in (3). See [11] for details of the PI method. In order to use the PI method to find a local minimizer of $\gamma(\mathbf{w})$, we need to shift $\gamma(\mathbf{w})$ by a *constant* scalar $c$, such that $\{\forall \mathbf{w} \in S^{d-1} : \gamma(\mathbf{w}) - c < 0\}$, where $S^{d-1}$ denotes the set of unit vectors $\mathbf{w} \in \mathbb{R}^d$. This way, all local minima of $\gamma(\mathbf{w})$ become local maxima of $|\gamma(\mathbf{w}) - c|$, so they can be found using a PI method of the form

$$\mathbf{w} \leftarrow \frac{[\mathbf{H}(\mathbf{w}) - c\mathbf{I}]\mathbf{w}}{\|[\mathbf{H}(\mathbf{w}) - c\mathbf{I}]\mathbf{w}\|}, \tag{18}$$

where the constant $c$ is obtained from the following Lemma.

**Lemma 3.** *Let $\mathbf{x}_1 \cdots, \mathbf{x}_n$ be a data set following the ICA model (1). Let $\mathbf{H}(\mathbf{w}) = \mathbb{E}_{F_n}\big[\frac{g(\mathbf{w}^\top \mathbf{x})}{\mathbf{w}^\top \mathbf{x}} \mathbf{x}\mathbf{x}^\top\big] \in \mathbb{R}^{d \times d}$ be positive*

*definite. Then* $\{\forall \mathbf{w} \in S^{d-1} : \gamma(\mathbf{w}) - c < 0\}$*, where* $c = \max\Big\{h(\mathbf{x}_i/\|\mathbf{x}_i\|),\ i \in \{1, \cdots, n\}\Big\}$,

*Proof.* Since any extremum of $\gamma(\mathbf{w})$ is an extremum of $h(\mathbf{w}) = \mathbb{E}[(\mathbf{w}^\top \mathbf{x})g(\mathbf{w}^\top \mathbf{x})]$, the problem changes to finding a constant $c$ such that $\{\forall \mathbf{w} \in S^{d-1} : h(\mathbf{w}) - c < 0\}$. Since $h(\mathbf{w}) \leq \max\Big\{(\mathbf{w}^\top \mathbf{x}_i)g(\mathbf{w}^\top \mathbf{x}_i),\ i \in \{1, \cdots, n\}\Big\}$, it is sufficient to show that for any data point $\mathbf{x}_i$, the vector $\mathbf{w}_i = \mathbf{x}_i/\|\mathbf{x}_i\|$ maximizes the Lagrangian:

$$\mathcal{L}(\mathbf{w}; \lambda) = (\mathbf{w}^\top \mathbf{x}_i)g(\mathbf{w}^\top \mathbf{x}_i) - \frac{\lambda}{2}(\mathbf{w}^\top \mathbf{w} - 1),$$

where $\lambda$ is the Lagrange multiplier. Setting the derivative of the Lagrangian w.r.t. $\mathbf{w}$ to zero gives

$$\mathbf{x}_i g(\mathbf{w}^\top \mathbf{x}_i) + \mathbf{x}_i \mathbf{w}^\top \mathbf{x}_i g'(\mathbf{w}^\top \mathbf{x}_i) - \lambda \mathbf{w} = \mathbf{0}, \qquad (19)$$

where $\lambda(\mathbf{w}) = \mathbf{w}^\top \mathbf{x}_i g(\mathbf{w}^\top \mathbf{x}_i) + (\mathbf{w}^\top \mathbf{x}_i)^2 g'(\mathbf{w}^\top \mathbf{x}_i)$ is obtained by multiplying both sides of (19) by $\mathbf{w}^\top$ from the left. Substituting $\lambda(\mathbf{w})$ in (19) and re-arranging the terms gives $\mathbf{w}_i = \mathbf{x}_i/(\mathbf{w}_i^\top \mathbf{x}_i)$, which holds true iff $\mathbf{w}_i = \mathbf{x}_i/\|\mathbf{x}_i\|$. □

We use $\delta(\mathbf{w})$ as a measure of non-Gaussianity to select the extracted component farther from Gaussianity. When more than one sources need to be extracted, we follow the same procedure as in the original k-unit *deflation-based* FastICA algorithm [2] but we use (17) and (18) instead of (4). The steps of such an algorithm are given in Algorithm 1, where $\Pi_{k-1}^{\perp}$ is an orthogonal projection operator that projects onto the orthogonal complement of the subspace (of the inner product space) spanned by the previously found FastICA demixing vectors $\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{k-1}$. Note that as shown in Algorithm 1, (17) and (18) can be run in parallel on two computing nodes. When only one node is available, the two loops need to be computed in series which takes more time.

**Algorithm 1:** PowerICA algorithm

**input** : $\mathbf{X} = \hat{\mathbf{D}}\mathbf{Y}$: Whitened data.
$(\mathbf{W}^{[0]})^\top = \begin{pmatrix}\mathbf{w}_1^{[0]} & \cdots & \mathbf{w}_d^{[0]}\end{pmatrix} \in \mathbb{R}^{d\times d}$: Random orthonormal matrix as initial guess.

**output** : $\hat{\mathbf{W}} = (\hat{\mathbf{w}}_1 \ \cdots \hat{\mathbf{w}}_d)^\top$: Demixing matrix estimate.

**for** $k = 1, \ldots, d-1$ **do**

1 $j \leftarrow 0$

2 $\Pi_{k-1}^{\perp} \leftarrow \mathbf{I} - \sum_{i=1}^{k-1} \hat{\mathbf{w}}_i \hat{\mathbf{w}}_i^\top$

| **repeat** **Node: 1** | **repeat** **Node: 2** |
|---|---|
| 3 $j \leftarrow j + 1$ | $j \leftarrow j + 1$ |
| 4 $\mathbf{w}_{k1}^{[j]} \leftarrow \mathbf{m}_{k1}^{[j-1]}$ | $\mathbf{w}_{k2}^{[j]} \leftarrow \mathbf{m}_{k2}^{[j-1]} - c\mathbf{w}_{k2}^{[j-1]}$ |
| 5 $\mathbf{w}_{k1}^{[j]} \leftarrow \Pi_{k-1}^{\perp}\mathbf{w}_{k1}^{[j]}$ | $\mathbf{w}_{k2}^{[j]} \leftarrow \Pi_{k-1}^{\perp}\mathbf{w}_{k2}^{[j]}$ |
| 6 $\mathbf{w}_{k1}^{[j]} \leftarrow \mathbf{w}_{k1}^{[j]}/\|\mathbf{w}_{k1}^{[j]}\|$ | $\mathbf{w}_{k2}^{[j]} \leftarrow \mathbf{w}_{k2}^{[j]}/\|\mathbf{w}_{k2}^{[j]}\|$ |
| **until** *convergence* | **until** *convergence* |

7 $\hat{\mathbf{w}}_k \leftarrow$ use $\delta(\mathbf{w})$ to choose between $\mathbf{w}_{k1}^{[j]}$ and $\mathbf{w}_{k2}^{[j]}$, i.e. the one with larger $\delta(\mathbf{w})$

8 $\hat{\mathbf{w}}_d \leftarrow \Pi_{d-1}^{\perp}\mathbf{w}_d^{[0]}/\|\Pi_{d-1}^{\perp}\mathbf{w}_d^{[0]}\|$

## VI. Numerical examples

We compare the convergence of the proposed PowerICA method with the deflation-based FastICA algorithm with nonlinearities $pow3$, $tanh$ and $gauss$. In our simulation setup, the data is a mixture of $d = 3$ sources possessing Laplacian, Uniform and Gaussian distribution with zero mean and unit variance. We use the same initial start $\mathbf{W}^{[0]}$ for both algorithms, where $\mathbf{W}^{[0]}$ is a random $d \times d$ matrix with elements from $\mathcal{N}(0, 1)$ distribution. Based on 1000 MC runs we report the number of non-convergent runs. We also compute the quality of the separation obtained by demixing matrix estimator $\hat{\mathbf{W}}$ using *interference to signal ratio* index

$$\text{ISR}(\hat{\mathbf{V}}) = \frac{1}{d(d-1)}\Big\{\sum_{i=1}^{d}\Big(\sum_{j=1}^{d}\frac{(\hat{v}_{ij})^2}{\max(\hat{v}_{i\cdot})^2} - 1\Big)\Big\}, \qquad (20)$$

where in the ideal case, matrix $\hat{\mathbf{V}} = \hat{\mathbf{W}}\mathbf{A}$ is a scaled and permuted copy of an identity matrix. Notation $\max(\hat{v}_{i\cdot})$ denotes the largest element in each row of $\hat{\mathbf{V}}$, which represents the signal power. The other elements of $\hat{\mathbf{V}}$ represent the interference power. The ISR obtains value in $[0, 1]$ where 0 implies perfect separation, whereas the maximal value 1 is pathological and obtained when $\hat{\mathbf{V}}$ is non-singular with $\hat{v}_{ij}$ equal in *each* row $i = 1, \ldots, d$. Recall that the asymptotic efficiency of the deflation-based FastICA estimates depends heavily on the order of extraction of the sources [12], [13]. Thus, in each MC run, we verify that the extraction order of sources is the same in both methods.

Table 1 illustrates the numerical stability of the proposed PowerICA algorithm. The "Failure" is defined as the number of non-convergent runs out of 1000 Monte Carlo trials. For example, in the case of $n = 20$ with $d = 3$ and nonlinearity $pow3$, the FastICA algorithm failed to converge 705 times out of 1000 trials. In contrast, PowerICA has always converged to a valid solution, i.e. ISR $< 0.22$, even with small number of samples, e.g. $n = 20$ and no failure occurs. The ISR value is averaged only over the cases that the FastICA algorithm converged. For PowerICA, the average ISR value over all 1000 MC runs is reported in parentheses.

TABLE I: Number of non-convergent runs and the average ISR values of the FastICA and the PowerICA algorithms for different values of $n$ and $d = 3$.

| | | FastICA | | | PowerICA | | |
|---|---|---|---|---|---|---|---|
| | $n$ | $pow3$ | $tanh$ | $gauss$ | $pow3$ | $tanh$ | $gauss$ |
| Fails | 20 | 705 | 589 | 597 | 0 | 0 | 0 |
| | 50 | 529 | 400 | 402 | 0 | 0 | 0 |
| | 100 | 294 | 216 | 212 | 0 | 0 | 0 |
| | 200 | 130 | 92 | 105 | 0 | 0 | 0 |
| ISR | 20 | 0.21 | 0.20 | 0.19 | 0.21(0.22) | 0.20(0.21) | 0.19(0.20) |
| | 50 | 0.11 | 0.09 | 0.10 | 0.11(0.13) | 0.09(0.11) | 0.10(0.11) |
| | 100 | 0.05 | 0.04 | 0.04 | 0.05(0.06) | 0.04(0.04) | 0.04(0.04) |
| | 200 | 0.02 | 0.01 | 0.01 | 0.02(0.02) | 0.01(0.01) | 0.01(0.01) |

## VII. Conclusion

In this letter, we provide an alternate derivation of the FastICA fixed-point algorithm. Note that FastICA algorithm was originally derived in [1], [2] as an approximate NR-update. Our derivation dose not need the unnecessary simplifying assumptions used in the original derivation. In addition, we propose a novel parallel ICA algorithm based on the power method. We showed that our proposed method is remarkably more stable than the FastICA algorithm and provides acceptable results event with small number of data points.

## REFERENCES


[1] A. Hyvärinen, "Fast and robust fixed-point algorithm for independent component analysis," *IEEE Trans. on Neural Networks*, vol. 10, pp. 626–634, 1999.

[2] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: John Wiley & Sons, 2001.

[3] T.-W. Lee, *Independent component analysis*. Springer, 1998.

[4] P. Tichavsky, Z. Koldovsky, and E. Oja, "Performance analysis of the FastICA algorithm and Cramér-Rao bounds for linear independent component analysis," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1189–1203, April 2006.

[5] J. Miettinen, K. Nordhausen, H. Oja, and S. Taskinen, "Deflation-based FastICA with adaptive choices of nonlinearities," *IEEE Transactions on Signal Processing*, vol. 62, no. 21, pp. 5716–5724, Nov 2014.

[6] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *The Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 124–127, 1950.

[7] H. Shen and K. Hüper, *On the Relationships Between Power Iteration, Inverse Iteration and FastICA*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 105–112.

[8] S. C. Douglas, *Relationships Between the FastICA Algorithm and the Rayleigh Quotient Iteration*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 781–789.

[9] J. Dennis Jr and R. Tapia, "Inverse, shifted inverse, and rayleigh quotient iteration as newtons method." [Online]. Available: http://www.caam.rice.edu/~rat/cv/RQI.pdf

[10] H. Benaroya, S. Han, and M. Nagurka, *Probability Models in Engineering and Science*, ser. McGraw-Hill professional engineering: Mechanical engineering. Taylor & Francis, 2005.

[11] G. H. Golub and H. A. van der Vorst, "Eigenvalue computation in the 20th century," *Journal of Computational and Applied Mathematics*, vol. 123, no. 12, pp. 35 – 65, 2000, numerical Analysis 2000. Vol. III: Linear Algebra.

[12] E. Ollila, "The deflation-based FastICA estimator: statistical analysis revisited," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, p. No.3, 2010.

[13] E. Ollila, H. J. Kim, and V. Koivunen, "Compact Cramér-Rao bound expression for independent component analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1421–1428, April 2008.