# Maximization of L1-norm Using Jacobi Rotations

Adam Borowicz
*Faculty of Computer Science, Department of Digital Media and Computer Graphics*
*Bialystok University of Technology*
Wiejska str. 45A, 15-351 Bialystok, Poland
a.borowicz@pb.edu.pl

***Abstract*—In recent years, we observe a growing interest in approaches to principal component analysis (PCA) based on L1-norm maximization. Unfortunately, existing L1-PCA algorithms are either computationally expensive or inaccurate. In this paper, we propose to use Jacobi-based rotational framework for solving L1-norm maximization problem. Under this framework, two new suboptimal algorithms are developed: the first one based on exhaustive angle search, and the second one based on a differentiable approximation of the absolute value function. Experimental studies show that the proposed approaches provide high accuracy as compared to the existing suboptimal algorithms. They are also considerably faster than currently the most accurate method based on bit-flipping. Simulation results show that both approaches can be used to perform independent component analysis (ICA) under whitening assumption achieving better robustness to outliers than other methods.**

***Index Terms*—L1-norm PCA, Jacobi rotations, ICA, outliers**

## I. Introduction

Principal component analysis (PCA) [1] is probably one of the most widely used techniques in multivariate signal processing. The PCA finds application in many areas, including machine learning, data compression, noise reduction, and statistical analysis. The major goal of the PCA is to identify orthogonal vectors that span the signal subspace, along which the data exhibit the greatest variability. Traditionally, this variability is measured using the Frobenius norm of the data matrix projected onto a given subspace.

Recently, a growing interest in approaches to the PCA based on the L1-norm maximization can be observed [2], [3], [4]. Unlike conventional methods, the L1-norm techniques offer an improved robustness to outliers i.e., data points that differ significantly from the other observations. Another attractive feature of the L1-PCA is that, when sources have negative kurtosis sign, it can perform independent component analysis (ICA) after data whitening [5]. Conventional PCA can be easily implemented using singular value decomposition (SVD) of the observation data matrix or eigen-decomposition (ED) of the sample covariance matrix. Unfortunately, there are no such simple solutions for the L1-PCA. Existing L1-PCA algorithms are either computationally intensive or inaccurate. Furthermore, the L1-norm maximization problem is not scalable. That is it can not be translated into series of the one-unit problems simply by projecting the data-matrix onto the null-space of the previous solution. Although the recently proposed suboptimal techniques [4], [6] provide significant computational improvements, their computational complexity increases quite fast with the number of data samples. Thus in the era of big data such algorithms can still be prohibitive for many applications.

This work was supported by Bialystok University of Technology under the grant WZ/WI-IIT/4/2020.

The Jacobi-based estimation framework is a well-known technique for diagonalizing symmetric matrices [7], [8]. It can also be used in data-driven algorithms [9], [10], [11] for the iterative transformations of multi-dimensional data through the plane rotations. These approaches are especially useful when the solution to the optimization problem is considered to be an orthogonal matrix. Namely, it is computed iteratively as a product of orthogonal matrices, specifically the Jacobi rotations. Although this framework does not guarantee the global convergence and depends on initialization, it is computationally efficient and easy to implement. In this study, we propose to use Jacobi-based optimization framework to solve L1-norm maximization problem. Two approaches are considered: the first one based on the exhaustive angle search algorithm, and the second one that uses a differentiable approximation of the absolute value function and simplified Newton iteration algorithm. Our experiments show superior computational efficiency and accuracy of the proposed algorithms, compared to the state-of-art techniques. By means of a numerical simulation, we also demonstrated that the proposed methods can be adapted to perform the ICA.

## II. L1-norm maximization

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ denotes data matrix of rank $d \leq \min\{D, N\}$, where $\{\mathbf{x}_i\}_{i=1}^N$ are the observation vectors. Then, the L1-norm maximization problem for $K \leq d$ can be formulated as follows:

$$\mathbf{Q}_{\text{L1}} = \underset{\substack{\mathbf{Q}=[\mathbf{q}_1,\ldots,\mathbf{q}_K]\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmax}} \sum_{i=1}^{K} \|\mathbf{X}^T\mathbf{q}_i\|_1, \tag{1}$$

where $\|.\|_1$ denotes L1-norm that return the sum of the absolute values of the individual entries. Since the absolute value function is non-differentiable, the problem is difficult to solve. However, in [3] it was shown that, if $\mathbf{X}\mathbf{B}_{\text{opt}} \overset{\text{SVD}}{=} \mathbf{U}\Sigma\mathbf{V}^T$, and

$$\mathbf{B}_{\text{opt}} = \underset{\mathbf{B}\in\{\pm 1\}^{N\times K}}{\operatorname{argmax}} \|\mathbf{X}\mathbf{B}\|_*, \tag{2}$$

where $\|.\|_*$ denotes nuclear norm, then $\mathbf{Q}_{\text{L1}} = \mathbf{U}\mathbf{V}^T$ is the optimal solution to (1). Therefore, the L1-norm maximization can be viewed as a combinatorial problem over

the binary field. In [3] exhaustive search algorithm with complexity $\mathcal{O}(N^{dK-K+1})$ has been proposed. A faster, yet suboptimal, version of this approach is based on consecutive bit-flipping operations [4]. Its time complexity is of order $\mathcal{O}(ND\min\{N,D\}+N^2(K^4+DK^2)+NDK^3)$, which can still be prohibitive for large data sizes. The most computationally efficient algorithm was proposed earlier in [2]. It is based on the following fixed-point (FP) iteration:

$$\mathbf{b}^{(t)} = \mathrm{sgn}(\mathbf{X}^T\mathbf{X}\mathbf{b}^{(t-1)}), \quad t = 2,3,..., \tag{3}$$

where $\mathbf{b}^{(1)} \in \{\pm 1\}^N$ is an initialization point. The principal vector is approximated by $\mathbf{q}_{\text{FP}} = \mathbf{X}\mathbf{b}_{\text{FP}}/\|\mathbf{X}\mathbf{b}_{\text{FP}}\|_2$, where $\mathbf{b}_{\text{FP}}$ is the converging point of the sequence $\{\mathbf{b}^{(t)}\}$. In order to compute $(K > 1)$ principal vectors, $\mathbf{X}$ is replaced by its projection onto the null-space of the previously calculated ones. The complexity of single iteration is of order $\mathcal{O}(NDK)$. However, the algorithm is rather inaccurate as it violates non-scalability principle of the L1-PCA problem.

## III. Proposed Methods

In the proposed optimization framework, the L1-norm metric is maximized by applying successively orthogonal transformations (rotations) to the data matrix. Let $p, q$ be two integers such that $1 \leq p < q \leq D$, then

$$\mathbf{G}(p,q,\theta) = \begin{bmatrix} \mathbf{I}_{p-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cos\theta & \mathbf{0} & \sin\theta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{q-p-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\sin\theta & \mathbf{0} & \cos\theta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{D-q-1} \end{bmatrix}, \tag{4}$$

represents Jacobi rotation [7] by the $\theta$ angle in the $(p,q)$ plane. The transformed data matrix at $t$-th rotation can be written as follows:

$$\mathbf{X}^{(t)} = \mathbf{G}(p_t,q_t,\theta_t)\mathbf{X}^{(t-1)}, \quad t = 2,3,..., \tag{5}$$

with $\mathbf{X}^{(1)} = \mathbf{W}\mathbf{X}$, where $\mathbf{W} \in \mathbb{R}^{D\times D}$ is an arbitrary orthonormal matrix that defines initialization point. All possible rotations represented by pairs $(p_t, q_t)$ are arranged in so-called sweeps. These sweeps are repeated cyclically until the maximum number of iterations is reached or when for all rotations in the current sweep we have $|\theta_t| < \epsilon$, where $\epsilon$ is a sufficiently small positive constant. Thus, the solution matrix $\mathbf{Q}_{\text{J}} \in \mathbb{R}^{D\times K}$, can be expressed as follows:

$$\mathbf{Q}_{\text{J}} = \mathbf{W}^T \left[ \overset{\curvearrowright}{\prod_t} \mathbf{G}(p_t,q_t,\theta_t)^T \right]_{*1:K}, \tag{6}$$

where $[\,.\,]_{*1:j}$ denotes first $j$ columns of an argument matrix. It is crucial for this framework to compute the rotation angles $\{\theta_t\}$ sequentially so that a given objective function is gradually optimized. Let denote by $\hat{x}_{ij}^{(t)}(\theta)$ the $(i,j)$-th entry of the data matrix (5) evaluated for the angle $\theta_t = \theta$. Then, the L1-norm objective function at $t$-th rotation is given by

$$J_{\text{L1}}^{(t)}(\theta) = \sum_{i=1}^{K}\sum_{j=1}^{N} |\hat{x}_{ij}^{(t)}(\theta)|. \tag{7}$$

Since the $t$-th rotation matrix modifies only $(p_t, q_t)$ rows of data matrix $\mathbf{X}^{(t-1)}$, the increase in objective function at $t$-th rotation can be written as follows:

$$J_{\text{L1}}^{(t)}(\theta) - J_{\text{L1}}^{(t-1)}(\theta_{t-1}) = \sum_{\substack{i\in\{p_t,q_t\} \\ i\leq K}} \sum_{j=1}^{N} |\hat{x}_{ij}^{(t)}(\theta)| - |x_{ij}|, \tag{8}$$

where $x_{ij}$ denotes the $(i,j)$-th entry of $\mathbf{X}^{(t-1)}$. It is clear that to get an increase in (7), it suffices to maximize only the first term in the above sum. In order to make this optimization framework more flexible, we introduce the following 'local' objective function:

$$J^{(t)}(\theta) = \begin{cases} J_{p_t}(\theta) + J_{q_t}(\theta) & \text{if } p_t \leq K \wedge q_t \leq K \\ J_{p_t}(\theta) & \text{if } p_t \leq K \wedge q_t > K \end{cases}, \tag{9}$$

where

$$J_i(\theta) = \sum_{j=1}^{N} f\left(\hat{x}_{ij}^{(t)}(\theta)\right), \quad i \in \{p_t, q_t\}, \tag{10}$$

with $f(x) \in \mathbb{R}$ being an arbitrary function of $x$. Please note that (9) is undefined for $p_t > K$, as the summation range in objective function (7) covers only indices from 1 to $K$. It means the rotations that would have to be performed entirely in the null-space can simply be omitted.

The first approach, we consider, can serve as a simple baseline algorithm for other approaches based on Jacobi rotations. Namely, we assume that $f(x) = |x|$ so that the maximization of (9) is equivalent to maximization of (7). Since $|x|$ is not differentiable, one-dimensional exhaustive search method is used to compute the objective function at $M$ equidistant points in some pre-defined interval. It can be shown that (9) is periodic with period $\pi$. Thus, our search for the optimal $\theta_t$ is restricted to the interval $[-\pi/2; \pi/2)$.

The second, less expensive approach is to use any differentiable approximation for the absolute value function. For example, the following function can be considered:

$$f(x) = \sqrt{x^2 + a}, \tag{11}$$

where $a$ is some small positive constant that controls the smoothness. In this case, a simplified Newton-Raphson procedure [10], [11] can be used to approximate rotation angle

$$\theta_t \approx \frac{\partial J^{(t)}(\theta=0)}{\partial\theta} \Big/ \left| \frac{\partial^2 J^{(t)}(\theta=0)}{\partial\theta^2} \right|. \tag{12}$$

The first- and second-order derivatives of (9) can be computed as follows:

$$\frac{\partial J_i(0)}{\partial\theta} = (2\delta_{ip_t} - 1)\sum_{j=1}^{N} f'(x_{ij})\, x_{i'j}, \tag{13}$$

$$\frac{\partial^2 J_i(0)}{\partial\theta^2} = \sum_{j=1}^{N} f''(x_{ij})\,(x_{i'j})^2 - f'(x_{ij})\, x_{ij}, \tag{14}$$

where $\delta_{ij}$ is a Kronecker delta, and $i' = \delta_{ip_t}q_t + \delta_{iq_t}p_t$. The terms $f'(x)$ and $f''(x)$ denote the first- and second-order derivatives of $f(x)$, respectively.

**Algorithm 1** Pseudocode of the Jacobi-based framework for maximization of the objective function $J(\theta, \ldots)$

**Require:**
$\mathbf{X} \in \mathbb{R}^{D \times N}, K \leq \text{rank}(\mathbf{X}), \mathbf{W} \in \mathbb{R}^{D \times D}, \mathbf{W}\mathbf{W}^T = \mathbf{I}_D$, $maxIter, \epsilon$
**Ensure:** $\mathbf{Q}_\text{J} \in \mathbb{R}^{D \times K}, \mathbf{Q}_\text{J}^T \mathbf{Q}_\text{J} = \mathbf{I}_K$

$\mathbf{X} \leftarrow \mathbf{W}\mathbf{X}$
**for** $k \in \{1, 2, \ldots, maxIter\}$ **do**
  $encore \leftarrow 0$
  **for** $p \in \{1, 2, \ldots, K\}$ **do**
    **for** $q \in \{p+1, p+2, \ldots, D\}$ **do**
      $\theta_\text{opt} \leftarrow \underset{-\pi/2 \leq \theta < \pi/2}{\text{argmax}} \; J(\theta, p, q, \mathbf{X})$
      **if** $|\theta_\text{opt}| > \epsilon$ **then**
        $encore \leftarrow 1$
        $\mathbf{X} \leftarrow \mathbf{G}(p, q, \theta_\text{opt})\mathbf{X}$
        $\mathbf{W} \leftarrow \mathbf{G}(p, q, \theta_\text{opt})\mathbf{W}$
  **if** $encore = 0$ **then**
    break
$\mathbf{Q}_\text{J} \leftarrow [\mathbf{W}^T]_{*1:K}$

Pseudocode of the Jacobi-based optimization framework is presented in Alg. 1, where we have tried to maximize readability, but not the computational efficiency. Please note that since the Jacobi rotation matrix $\mathbf{G}$ modifies only $(p, q)$ rows, it is not necessary to compute it explicitly. In order to generate the pairs $(p, q)$, a row-cycling ordering was used, but other arrangements are also possible. It is easy to see that in each iteration/sweep, we must perform up to $K(K-1)/2 + (D-K)K$ data rotations. Unfortunately, in order to perform data rotation, we also need to estimate the optimal rotation angle which is usually more demanding than data rotation itself. However, both operations, i.e., the transformation of the data matrix $\mathbf{X}$, and the computation of the rotation angle depend linearly on the number of data points $N$. Taking into account that the exhaustive search algorithm must evaluate the objective function at $M$ points, time complexity of the single sweep can be roughly estimated as of order $\mathcal{O}(MNDK)$. In the case of the method based on a differentiable approximation of $|x|$, the parameter $M$ should be dropped as the derivatives (14) are calculated at point $\theta = 0$, only.

## IV. Experimental studies

The proposed Jacobi-based methods have been implemented in the MATLAB environment. For convenience, the exhaustive angle search algorithm is denoted as L1-JEX, while the algorithm based on the differentiable approximation is denoted as L1-JDA. It is rather common that the performance of an iterative algorithm may vary depending on the stop conditions and initialization. Therefore, for both approaches, in all experiments, the maximum number of iterations was limited to 100 and the precision parameter $\epsilon$ was set to $\pi/(M = 512)$. We found empirically that the proposed methods perform well when they are initialized using the eigenvectors of the matrix $\mathbf{X}\mathbf{X}^T$. A similar approach was suggested in [4] for the L1-BF algorithm, where SV-sign initialization was used.

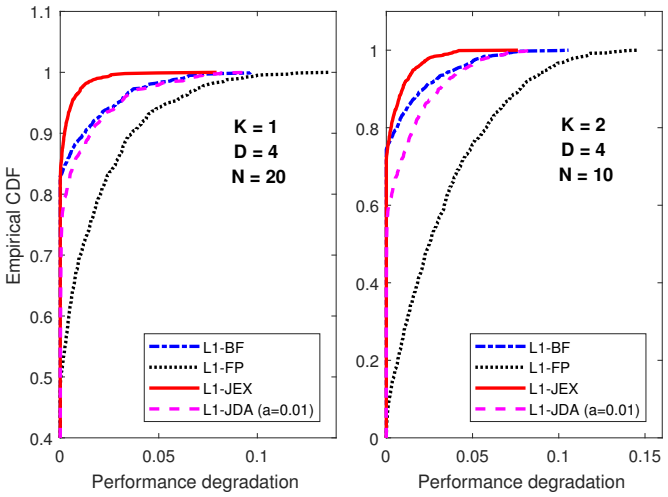


Fig. 1: Empirical CDF of performance degradation ratio estimated for various L1-PCA algorithms

### A. Performance degradation

In order to measure the performance degradation attained by the proposed algorithms, we followed a similar procedure to that in [4]. Two scenarios were considered: the first one with $D = 4$, $N = 20$, $K = 1$, and the second one with $D = 4$, $N = 10$, $K = 2$. In both scenarios, we generated 1000 random data matrices with entries drawn from a Gaussian distribution $\mathcal{N}(0, 1)$. For each data matrix, the orthonormal matrix $\mathbf{Q}$ was computed by using the state-of-art methods, i.e., L1-BF [4], L1-FP [2] and proposed ones. The performance degradation experienced by this matrix on the L1-norm objective function was measured as follows:

$$\Delta(\mathbf{Q}, \mathbf{X}) = \frac{\|\mathbf{X}^T \mathbf{Q}_\text{L1}\|_1 - \|\mathbf{X}^T \mathbf{Q}\|_1}{\|\mathbf{X}^T \mathbf{Q}_\text{L1}\|_1}, \tag{15}$$

where $\mathbf{Q}_\text{L1}$ denotes the matrix obtained by the optimal L1-PCA algorithm [3], for the same data. Fig. 1 presents the empirical cumulative distribution functions (ECDFs) of (15) estimated as the fractions of the measurements that are less than or equal to the specified values. In the first scenario $(K = 1)$, both the L1-JEX and L1-BF methods return the exact solution in about 84 percent of runs. However, the performance degradation attained by the L1-JEX method is, with empirical probability 1, less than 0.07, which is the best result among all methods. As expected, the L1-JDA method is slightly less accurate than the L1-BF algorithm, but it is superior to the L1-FP method. In the second scenario $(K = 2)$, the performance degradation is more prominent for all methods, but once again the L1-JEX algorithm achieves higher values of the metric (15) more frequently than any other method. The most significant performance loss can be seen for the L1-FP method. Please note that the obtained results are consistent with those in [4]. The small differences are due to the slightly different data sizes used in this experiment.

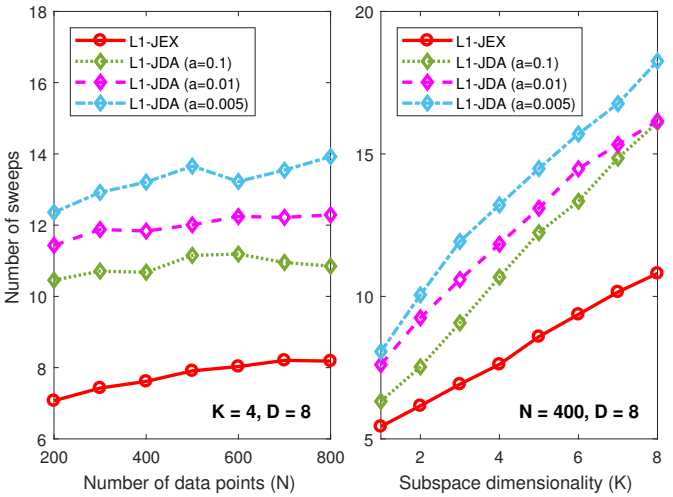


Fig. 2: Average number of sweeps taken by proposed methods until convergence vs number of observations (left) and subspace dimensionality (right).

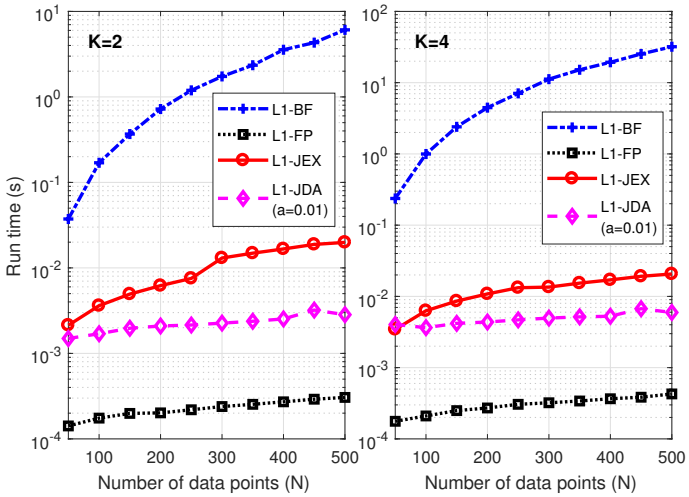


Fig. 3: Average execution times vs number of observations measured for proposed methods and state-of-art suboptimal algorithms

### *B. Convergence and execution time*

Since the sweeps are repeated until convergence is achieved, the total execution time depends on the convergence rate. A rigorous convergence analysis of the proposed methods is not an easy task and is out of the scope of this study. However, we measured the average number of iterations taken by the proposed algorithms until convergence was reached. The results averaged over 1000 independent runs are presented in Fig. 2. Average number of iterations shows a weaker than linear dependence on the number of data points, but there is a strong linear dependence on the subspace dimensionality. In the case of the L1-JDA method, it can also be observed that the number of iterations increases as the parameter $a$ decreases. It is rather not surprising, since this parameter affects the smoothness of the objective function and, in turn, the convergence rate of the Newton method. The L1-JDA approach requires more iterations to converge than the L1-JEX method, but as will be shown in the next experiment, these iterations are less computationally expensive. We also observed that the L1-JEX method always converged to stationary solution within the iteration limit, while the L1-JDA method reached this limit in around 5 percent of the runs.

In order to compare the computational performance of the proposed algorithms with the state-of-art suboptimal methods [2], [4], we measured their execution times for various data sizes. The experiments were carried out in MATLAB environment running on AMD Ryzen 5 3550H processor. Fig. 3 presents the execution times obtained for all methods averaged over 100 independent runs. It can be seen that for $K > 1$ and large $N$, the proposed methods are much faster than the L1-BF algorithm. Their execution times increase linearly with the number of data points. This is serious improvement compared to the L1-BF algorithm, where the execution time increases quadratically with $N$. The L1-JDA method is slightly faster than the L1-JEX method, yet it can not compete with the L1-FP method, which turns out to be the fastest suboptimal approach.

### *C. ICA using L1-PCA algorithms*

As shown in [5], the ICA can actually be performed by L1-PCA after L2-norm whitening. However, when the sources have positive kurtosis, L1-PCA algorithms should be modified to perform L1-norm minimization instead of maximization. Due to high computational demands in this experiment, we consider only sub-optimal algorithms. Please note that, for $K = 1$, they can be efficiently implemented in deflation-based framework [5], where the ICs are extracted one-by-one. On the other hand, the results presented in the previous section show that the proposed methods are also able to estimate jointly $(K > 1)$ L1-norm components within a reasonably short execution time. Therefore, we compared an independent source extraction performance of the proposed methods, for $(K = D = 8)$ with that of the existing L1-PCA algorithms (implemented on a deflationary basis). For comparative purposes also FastICA algorithm [12] with the fourth-power nonlinearity was implemented as a baseline ICA algorithm. We considered the synthetic sources all with negative or positive kurtosis sign generated from uniform and Laplacian distribution, respectively. In order to evaluate the robustness of the algorithms against outliers, in one scenario, 2% of randomly chosen observations were replaced with noise spikes drawn from a Gaussian distribution $\mathcal{N}(3, 1)$. In the case of the Laplacian distribution, all L1-PCA algorithms except the L1-FP, were modified as suggested in [5]. For example, in order to perform L1-norm minimization using the L1-JDA method, it suffices to change sign of the rotation angle in (12). The separation quality was estimated using the signal-to-interference ratio (SIR) [11] averaged over all components. Also, execution times were measured under the same stop conditions as in the previous experiments. All performance

TABLE I: Independent sources extraction performance $(D = 8)$.

| | SIR (dB) | | | | | | Run time (ms) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Distribution | uniform | | | Laplacian | | | uniform | | | Laplacian | | |
| $N$ | 200 | 400 | 800 | 200 | 400 | 800 | 200 | 400 | 800 | 200 | 400 | 800 |
| FastICA-4pow | **15.69** | **19.64** | **23.16** | 8.33 | 11.09 | 14.31 | 4 | **2** | **3** | 7 | **3** | **4** |
| L1-FP | 4.62 | 10.06 | 15.50 | -4.86 | -5.35 | -5.54 | **2** | 4 | 9 | **2** | **3** | 9 |
| L1-BF | 6.16 | 11.31 | 15.74 | -2.50 | -2.60 | -2.42 | 59 | 197 | 1007 | 50 | 114 | 513 |
| L1-JEX | 12.05 | 16.86 | 19.82 | 13.89 | 18.43 | **21.95** | 50 | 129 | 282 | 66 | 94 | 242 |
| L1-JDA $(a = 0.1)$ | 13.08 | 17.44 | 20.79 | **14.72** | **18.60** | 21.82 | 22 | 29 | 42 | 44 | 40 | 81 |

TABLE II: Independent sources extraction performance for data contaminated by outliers $(D = 8)$.

| | SIR (dB) | | | | | | Run time (ms) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Distribution | uniform | | | Laplacian | | | uniform | | | Laplacian | | |
| $N$ | 200 | 400 | 800 | 200 | 400 | 800 | 200 | 400 | 800 | 200 | 400 | 800 |
| FastICA-4pow | 0.71 | -2.47 | -1.09 | 4.09 | 4.69 | 7.31 | 10 | 15 | 17 | 7 | **3** | **5** |
| L1-FP | 0.14 | 0.73 | 3.89 | -4.29 | -4.70 | -5.36 | **2** | **5** | **8** | **2** | **3** | 9 |
| L1-BF | 1.08 | 1.23 | 4.07 | -2.61 | -2.21 | -2.44 | 68 | 294 | 1277 | 47 | 112 | 508 |
| L1-JEX | 4.96 | 7.55 | 11.53 | 10.61 | 13.77 | **18.13** | 53 | 192 | 356 | 71 | 117 | 288 |
| L1-JDA $(a = 0.1)$ | **5.03** | **7.60** | **12.08** | **11.05** | **13.78** | 18.07 | 22 | 47 | 60 | 53 | 49 | 80 |

metrics were averaged over 100 random realizations of the sources and the mixing matrices. The results are presented in Tab. I-II, where the bold fonts indicate the best performance. The FastICA provides the best source separation quality for uniform sources, but it does not perform well for Laplacians. As opposed to the L1-PCA algorithms, it is also very sensitive to outliers. The proposed methods attain relatively high SIRs for uniform sources and outperform other L1-PCA algorithms under all experimental conditions. The L1-JDA method provides shorter execution times than the L1-JEX. Surprisingly, there is no clear difference in separation quality between these methods. Although the proposed methods are not as fast as the L1-FP or FastICA algorithm, they are faster and more accurate than the L1-BF algorithm for large $N$. As can also be observed, the performance of the L1-BF method for Laplacian distribution is very poor. The L1-BF algorithm is based on the maximization of the nuclear norm over binary field, which is equivalent to maximization of the L1-norm in $\mathbb{R}^D$. Since the L1-norm and the nuclear norm minimization problems are not related in the same way, the modification proposed in [5] may not work well in this case. On the other hand, the proposed methods try to minimize the L1-norm explicitly in $\mathbb{R}^D$, which results in superior source extraction quality.

## V. Conclusions

We found that Jacobi-based optimization framework is well suited for solving the L1-norm maximization problems. Under this framework, two novel L1-PCA algorithms have been proposed. The first one, abbreviated as L1-JEX, performs exhaustive angle search for each rotation plane. The second one, abbreviated as L1-JDA, uses a differentiable approximation for absolute value function and calculates the rotation angles using the simplified Newton method. Experimental studies show a superior accuracy of the L1-JEX method, compared to the existing algorithms. As compared to the L1-JEX method, the L1-JDA provides some computational improvements at the expense of higher L1-norm performance degradation. Nevertheless, both methods are considerably faster than popular bit-flipping algorithm, especially for large data sizes. In view of recent discovery on the link between L1-PCA and ICA, it was demonstrated that the proposed L1-PCA algorithms can perform ICA under the whitening assumption, and offer better robustness to outliers than other methods.

## References


[1] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer Verlag, 2002.

[2] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1672–1680, 2008.

[3] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for $L_1$-subspace signal processing," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 5046–5058, 2014.

[4] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L1-norm principal-component analysis via bit flipping," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4252–4264, 2017.

[5] R. Martın-Clemente and V. Zarzoso, "On the link between L1-PCA and ICA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 515–528, 2017.

[6] M. Dhanaraj and P. P. Markopoulos, "Novel algorithm for incremental L1-norm principal-component analysis," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 2020–2024.

[7] G. Golub and C. Van Loan, *Matrix Computations*. USA: Johns Hopkins University Press, 2013.

[8] J. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 6, pp. 362–370, 1993.

[9] J. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999.

[10] W. Ouedraogo, A. Souloumiac, and C. Jutten, "Non-negative independent component analysis algorithm based on 2D Givens rotations and a Newton optimization," in *Latent Variable Analysis and Signal Separation*. Berlin, Heidelberg: Springer, 2010, pp. 522–529.

[11] A. Borowicz, "Orthogonal approach to independent component analysis using quaternionic factorization," *EURASIP Journal on Advances in Signal Processing*, vol. 2020, no. 39, p. 23, September 2020.

[12] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 626–634, 1999.