Signal Processing 189 (2021) 108286

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

FFT calculation of the L1-norm principal component of a data matrix

Stefania Colonnese [a,*], Panos P. Markopoulos [b,1], Gaetano Scarano [a], Dimitris A. Pados [c]

[a] Department of Information Engineering, Electronics, and Telecommunications, University of Rome "Sapienza", Rome, Italy
[b] Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY, USA
[c] Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA

ARTICLE INFO

Article history:
Received 23 December 2020
Revised 6 August 2021
Accepted 8 August 2021
Available online 10 August 2021

Keywords:
L1-norm
L1-PCA
Fourier transform
FFT
Cyclic shift

ABSTRACT

This paper presents a fast approximate rank-1 L1-norm Principal Component Analysis (L1-PCA) estimator implemented in the Fourier domain. Specifically, we first rephrase the problem of rank-1 L1-PCA estimation as a cyclic shift parameter estimation and then we present an algorithm for estimating the first L1-norm Principal Component (L1-PC) in the Fourier domain, practically using FFT. The proposed method is shown to be asymptotically efficient and our numerical studies corroborate its performance merits.

1. Introduction

L1-norm Principal Component Analysis (L1-PCA) is a fundamental tool in robust data analysis, and it has recently found application to big data problems such as unsupervised classification [1] or linear discriminative analysis [2]. The need for efficient L1-PCA computation has solicited recent research for fast convergence iterative solutions leveraging, among other techniques, binary data weighting [3,4], Grassmann average [5,6], Grassmann manifold optimization [7].

Here, we present a novel approach to L1-PCA estimation in the Fourier domain. Specifically, we express the L1-PCA formulation as a binary weighted combination of observations and we rephrase the problem of rank-1 L1-PCA as a phase (i.e., cyclic shift) estimation problem. Then, we solve this problem in the Fourier domain, using FFT so as to reduce computational complexity on big data samples. Our numerical studies show that the resulting algorithm maintains the intrinsic outlier resilience of L1-PCA while being computationally much lighter than alternatives.

The structure of the paper is as follows. In Sections 2 and 3.2 we present the L1-PCA formulation and the proposed FFT-based computation, respectively. In Section 4, we assess the performance of the proposed methods in multiple numerical studies and compare it to alternative techniques in the literature. Finally, in Section 5, we draw some conclusions.

2. Rank-1 L1-PCA

Let us consider the $D \times N$ data matrix $\mathbf{X}$ whose columns $\mathbf{x}_i, i \in [N] = \{1, 2, \ldots, N\}$, comprise N measurements in $\mathbb{R}^D$, where $N > D$. The L1-PC of $\mathbf{X}$ is defined as

$$\mathbf{q}_{L1} = \underset{\substack{\mathbf{q} \in \mathbb{R}^D \\ \|\mathbf{q}\|_2 = 1}}{\operatorname{argmax}} \|\mathbf{X}^\top \mathbf{q}\|_1. \tag{1}$$

In [3], authors showed that

$$\mathbf{q}_{L1} = \mathbf{X}\mathbf{b}_{L1} \|\mathbf{X}\mathbf{b}_{L1}\|_2^{-1} \tag{2}$$

is a solution to (1), when $\mathbf{b}_{L1}$ is a solution to

$$\max_{\mathbf{b} \in \{\pm 1\}^N} \|\mathbf{X}\mathbf{b}\|_2. \tag{3}$$

For the formulation in (3), [3] proposed an efficient bit-flipping based solver.

The role of the optimal binary vector $\mathbf{b}_{L1}$ is illustrated in Fig. 1, scatter-plotting the points of the data matrix $\mathbf{X}$, drawn from multivariate Gaussian distribution ($N = 512$, $D = 2$). The color of each point $\mathbf{x}_i$ represents its optimal binary weight (± 1 sign) in $[\mathbf{b}_{L1}]_i$. The above observation highlights the intrinsic unsupervised binary classification property of L1-PCA, which is leveraged in Martín-Clemente and Zarzoso [1].

From Fig. 1, a further observation stems. It is clear that the role of the optimal signs $\mathbf{b}_{L1}$ is to apply a π-shift to the phase of a few

* Corresponding author.
E-mail address: stefania.colonnese@uniroma1.it (S. Colonnese).
[1] Eurasip Member.

https://doi.org/10.1016/j.sigpro.2021.108286
0165-1684/

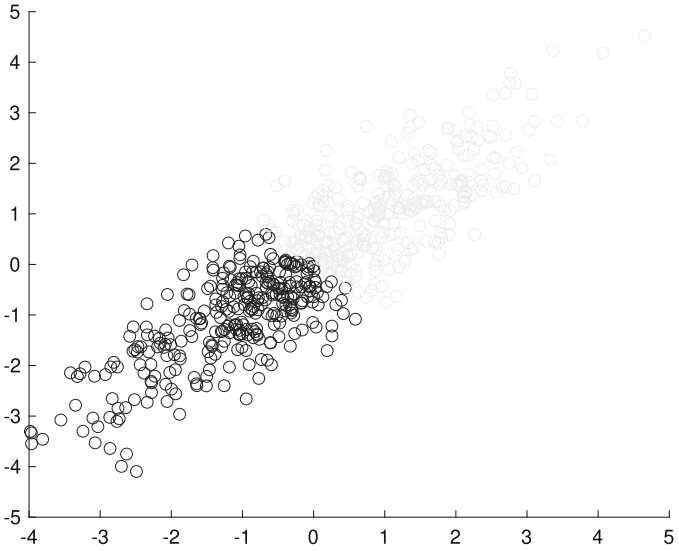


Fig. 1. Scatter-plot of data matrix $\mathbf{X}$ drawn from multivariate Gaussian distribution ($N = 512$, $D = 2$). The color of each point $\mathbf{x}_i$ represents the corresponding optimal sign in $[\mathbf{b}_{L1}]_i$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

measurements so as to map them into the half-plane where they add constructively, thus providing the maximum euclidean norm $\|\mathbf{X}\mathbf{b}_{L1}\|_2$. Inspired by this observation, we rephrase the search for the L1-PC as a search for the phase of the half-plane into which the measurements, either unchanged or phase-shifted by a π-phase, additively combine into a vector of maximum L2 norm.

3. Proposed Fourier-domain rank-1 L1-PCA

3.1. L1-PCA as an optimal half-plane search

For simplicity, we first study $D = 2$. We observe that the unit vector $\mathbf{q}_{L1}$ assumes directions belonging to the set

$$\Theta = \left\{\theta :\ \theta = \arg(\mathbf{X}\mathbf{b}),\ \mathbf{b} \in \{\pm 1\}^N\right\}, \tag{4}$$

where $\arg(\cdot):\ \mathbb{R}^2 \to (0, 2\pi]$ returns the angle of its argument with the positive horizontal axis. We note that Θ is a finite set with cardinality $|\Theta| \leq 2^N$. Accordingly, the maximization in (1) can be rewritten as

$$\max_{\theta\in\Theta} \|\mathbf{X}^\top \mathbf{u}(\theta)\|_1 \tag{5}$$

where $\mathbf{u}(\theta) = [\cos(\theta), \sin(\theta)]^\top$, so that $\arg(\mathbf{u}(\theta)) = \theta$ and $\|\mathbf{u}(\theta)\|_2 = 1$. Noticing that $|a| = \max_{b\in\{\pm 1\}} ba$, the above problem can be rewritten as

$$\max_{\theta\in\Theta} \max_{\mathbf{b}\in\{\pm 1\}^N} \sum_{i\in[N]} b_i \mathbf{x}_i^\top [\cos(\theta), \sin(\theta)]^\top. \tag{6}$$

Setting $z_i = (b_i + 1)/2$ and $\phi_i = \arg(\mathbf{x}_i)$, (6) becomes

$$\max_{\theta\in\Theta, \mathbf{z}\in\{0,1\}^N} \sum_{i\in[N]} \cos(z_i\pi) \|\mathbf{x}_i\|_2 \mathbf{u}(\phi_i)^\top \mathbf{u}(\theta) \tag{7}$$

$$= \max_{\theta\in\Theta, \mathbf{z}\in\{0,1\}^N} \sum_{i\in[N]} \|\mathbf{x}_i\|_2 \cos(z_i\pi)[\cos(\phi_i), \sin(\phi_i)]\cdot \cdot [\cos(\theta), \sin(\theta)]^\top \tag{8}$$

$$= \max_{\theta\in\Theta, \mathbf{z}\in\{0,1\}^N} \sum_{i\in[N]} \|\mathbf{x}_i\|_2 \cos(\phi_i - \theta)\cos(z_i\pi). \tag{9}$$

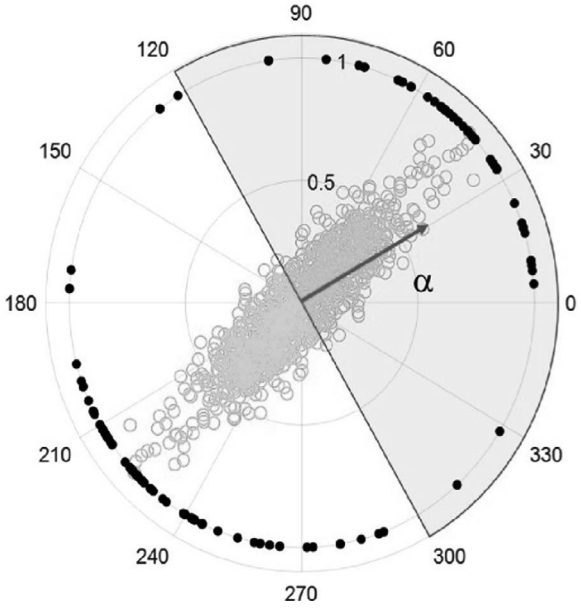


Fig. 2. Scatter-plot of Gaussian data $\mathbf{X}$ (green points), examples of discrete directions in Θ (black points), α-dependent half-plane (shaded light blue area) ($N = 512$, $D = 2$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Next, we introduce function $\beta : \mathbb{R}^2 \to \{0, \pi\}$

$$\beta(\alpha_1, \alpha_2) = \begin{cases} \pi, & \text{for } |\alpha_1 - \alpha_2| \geq \pi/2 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

which absorbs the maximization over $\mathbf{z}$ so that (9) becomes

$$\max_{\theta\in\Theta} \sum_{i\in[N]} \|\mathbf{x}_i\|_2 \cos(\phi_i - \theta)\cos(\beta(\phi_i, \theta)) \tag{11}$$

$$= \max_{\theta\in\Theta} \sum_{i\in[N]} \|\mathbf{x}_i\|_2 \cos(\phi_i + \beta(\phi_i, \theta) - \theta) \tag{12}$$

$$= \max_{\theta\in\Theta} \sum_{i\in[N]} \|\mathbf{x}_i\|_2 (\cos(\phi_i + \beta(\phi_i, \theta))\cos(\theta) + + \sin(\phi_i + \beta(\phi_i, \theta))\sin(\theta)) \tag{13}$$

$$= \max_{\theta\in\Theta} \sum_{i\in[N]} \|\mathbf{x}_i\|_2 \mathbf{u}(\phi_i + \beta(\phi_i, \theta))^\top \mathbf{u}(\theta). \tag{14}$$

Next, we define

$$\mathbf{m}(\theta) = \sum_{i\in[N]} \|\mathbf{x}_i\|_2 \mathbf{u}(\phi_i + \beta(\phi_i, \theta)) \tag{15}$$

We notice that we obtain this vector if we add our data points, after tuning their phases (adding π or nothing) so that each one forms an acute angle with $\mathbf{u}(\theta)$. Accordingly, we define $M(\theta) = \|\mathbf{m}(\theta)\|_2$, and $\Phi(\theta) = \arg(\mathbf{m}(\theta))$ –both piecewise constant functions whose transitions are determined by the data point phases $\{\phi_i\}_{i=1}^N$. Next, we notice that (14) can be rewritten as

$$\max_{\theta\in\Theta} M(\theta)\mathbf{u}(\Phi(\theta))^\top \mathbf{u}(\theta). \tag{16}$$

That is, if θ_{opt} is a solution to (16), then $\mathbf{q}_{L1} = \mathbf{u}(\theta_{opt})$ is the L1-PC in (1).

The above maximization is carried out with respect to $\theta \in \Theta$. This is exemplified in Fig. 2 ($N = 512$, $D = 2$), where we present the scatter-plot of Gaussian data $\mathbf{X}$ (green points), and a few examples of the discrete directions in Θ (black points). Adding the

data points, unaltered or flipped so as to be tilted towards θ, we obtain the vector $\mathbf{m}(\theta)$. The problem in (16) consists in finding a direction $\theta \in \Theta$ (i.e. a black point in Fig. 2) such that the module of $\mathbf{m}(\theta)$ is maximum at θ_{opt}. This condition is achieved when the data measurements are phase-shifted by $\beta(\phi_i, \theta_{opt})$ so as to add constructively in the half-plane centered in θ_{opt}[2] Therefore, seeking for θ_{opt} equals to seeking for a half-plane such that the phase-shifted by $\beta(\phi_i, \theta_{opt})$ add constructively.. The problem in (16) can then be solved by first finding the half-plane in the direction α in such that:

$$\alpha_{opt} = \underset{\alpha\in[0,2\pi)}{\operatorname{argmax}} \; M(\alpha) \tag{17}$$

and then returning

$$\theta_{opt} = \Phi(\alpha_{opt}). \tag{18}$$

and then computing the related θ as the phase $\Phi(\alpha_{opt})$ of their algebraic sum. By doing this, the maximization of $\mathbf{m}(\theta)$ over the discrete set Θ is expressed in terms of the maximization of the piecewise constant function $M(\alpha)$.

To recap, in the above analysis we firstly start from the search for the maximum of (16) with θ varying in the discrete set Θ, exemplified by the black points in Fig. 2. Then, we replace it with the search for the maximum of $M(\alpha)$ with α varying in the continuous interval $[0, \pi)$; this corresponds to maximize $M(\alpha)$ by varying the orientation of the α dependent half-plane (i.e. the blue shaded area in Fig. 2). Finally, $\theta_{opt} = \Phi(\alpha_{opt})$ is computed; remarkably, since $M(\alpha)$ is a piecewise constant function, different values of α_{opt} can be found; still, all these values lead to the same θ_{opt}. Thereby, we have managed to rephrase rank-1 L1-PCA as a search for the phase α_{opt} of the half-plane into which the measurements, either unchanged or phase-shifted by a π, combine to a maximum L2 norm vector. The optimal phase shifts $\beta\left(\phi_i, \alpha_{opt}\right)$ play the same role as the optimal signs $\mathbf{b}_{L1}$: they apply a π-shift to the phase of a few measurements. By doing this, the data are mapped into the half-plane where all the measured points (either in their original or flipped version) add constructively.

3.2. Fourier-domain computation

Based on the above analysis, we can now design an FFT-based solver for L1-PCA. To that end, we will transfer our notation from $\mathbb{R}^2$ to $\mathbb{C}$. Accordingly, data point $\mathbf{x}_i \in \mathbb{R}^2$ can be represented by $m_i e^{j\phi_i}$, where $m_i = \|\mathbf{x}_i\|_2$ and $\phi_i = \arg(\mathbf{x}_i)$. Similarly, the metric in (17) becomes $M(\alpha) = |\mathcal{M}(\alpha)|$, where

$$\mathcal{M}(\alpha) = \sum_{i\in[N]} m_i e^{j\xi_i(\alpha)} \tag{19}$$

and

$$\xi_i(\alpha) = \begin{cases} \phi_i, & \text{if } |\alpha - \phi_i| \le \pi/2 \\ \phi_i + \pi, & \text{otherwise} \end{cases} \tag{20}$$

Next, we proceed with estimating the maximizer of $M(\alpha)$, α_{opt}, as follows. First, we partition the range $[0, 2\pi)$ into K intervals/sectors of equal width $2\pi/K$ rads,

$$\Delta(k) = \left[(k-1)\frac{2\pi}{K},\; k\frac{2\pi}{K}\right), \quad k \in [K]. \tag{21}$$

For every k, $\Delta(k)$ is centered around

$$\delta_k = \frac{2\pi}{K}(k - 1/2). \tag{22}$$

The result of the partition is summarized in Fig. 3 (left), that shows the angular intervals $\{\Delta(k)\}_{k\in[K]}$, for $K = 64$ (blue sectors).

[2] This is demonstrated by contradiction. In fact, should any of the measurement be out of the half-plane centered in θ_{opt}, is phase-shift by π would lead to an increase in $\mathbf{m}(\theta)$ with respect to $\mathbf{m}(\theta_{opt})$, which contradicts the definition of θ_{opt}.

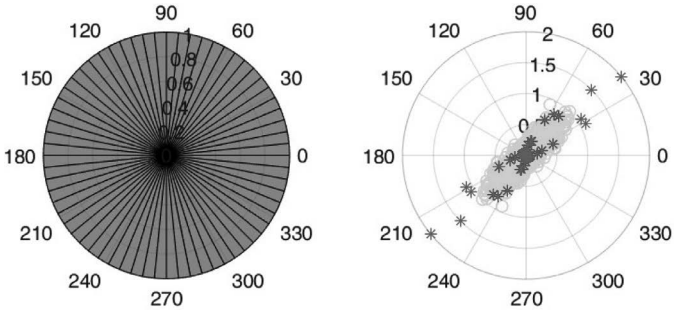


Fig. 3. Left: Partition of $[0, 2\pi)$ in K sectors $\{\Delta(k)\}_{k\in[K]}$. Right: Original points $m_i e^{j\phi_i}$ (green circles) and values $z_k e^{j\delta_k}$ (red circles). Data matrix $\mathbf{X}$ drawn from multivariate Gaussian distribution ($N = 512$, $D = 2$, $K = 64$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Given this partition, the data, exemplified by the green circles in Fig. 3 (right), can be assigned to different intervals as follows. For every $k \in [K]$, we define

$$\mathcal{I}_k = \{i \in [N] : \phi_i \in \Delta(k) \text{ or } \phi_i + \pi \in \Delta(k)\}. \tag{23}$$

That is, $\mathcal{I}_k$ contains the indices of data points that belong to $\Delta(k)$ or are therein projected by a π-shift. For the sake of completeness, we denote by z_k the accumulation of the magnitudes of all data points with index in $\mathcal{I}_k$, namely $z_k = \sum_{i\in\mathcal{I}_k} m_i$. The values $z_k e^{j\delta_k}$ are represented by the red points in Fig. 3(right). These values will be leveraged as a proxy of the original data in the following computation.

Let us consider $\mathbf{w}(\alpha) \in \mathbb{R}^K$ such that

$$w_k(\alpha) = \begin{cases} 1, & \alpha - \pi/2 < (k-1/2)\frac{2\pi}{K} \le \alpha + \pi/2 \\ 0, & \text{otherwise} \end{cases}. \tag{24}$$

That is, $\mathbf{w}(\alpha)$ represents a window function centered around an index depending on α and K. As α varies, the window of $\mathbf{w}(\alpha)$ shifts over different indexes k, covering different phase sectors $\Delta(k)$. This operation represents the counterpart, in the discrete domain, of sliding the α-dependent half-plane addressed in (17).

A visual summary of the variables involved in the above analysis is given in Fig. 4. Fig. 4(left) plots the partition into angular intervals and the values $z_k e^{j\delta_k}$ (red circles) computed accordingly. Besides, the α-dependent half-plane is represented in shaded light blue. From the definition in (24), we recognize that the kth angular interval is associated to a value $w_k(\alpha)$ depending on the plane orientation. In Fig. 4(right) we plot z_k (red samples) and $w_k(\alpha)$ (blue samples), i.e. the entries of $\mathbf{z}$ and $\mathbf{w}(\alpha)$, as discrete functions of the index k ($N = 512$, $D = 2$, $K = 64$). This representation suggests that the search for the optimal half-plane in Fig. 4(left) can be expressed as the search for an optimal window shift in Fig. 4(right). This intuition motivates the derivation of the FFT-based PCA computation detailed in the following.

Based on the above definitions, for any $\alpha \in [0, 2\pi)$, the sum in (19) can be rewritten by reordering the terms in groups of points belonging to different phase intervals. Then we obtain:

$$\mathcal{M}(\alpha) = \sum_{k\in[K]} w_k(\alpha) \sum_{i:\; \xi_i(\alpha)\in\Delta(k)} m_i e^{j\xi_i(\alpha)}. \tag{25}$$

Importantly, we notice that as K increases asymptotically, for every $k \in [K]$, $\Delta(k)$ contracts around its center, δ_k –i.e., every $x \in \Delta(k)$ is asymptotically near δ_k. Accordingly, for large enough K, any α, and i such that $\xi_i(\alpha) \in \Delta(k)$, we can approximate $\xi_i(\alpha)$ by δ_k.

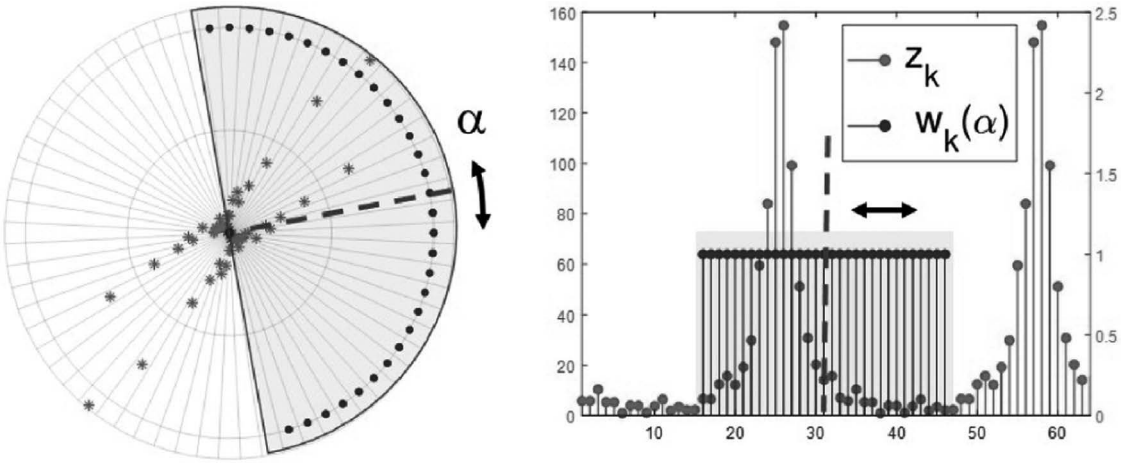


Fig. 4. Left: plot of the values $z_k e^{j\delta_k}$ (red circles), and α-dependent half-plane (shaded light blue area). Right: plot of the entries of $\mathbf{z}$ and $\mathbf{w}(\alpha)$. Data matrix $\mathbf{X}$ drawn from multivariate Gaussian distribution ($N = 512$, $D = 2$, $K = 64$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Accordingly, we estimate (25) as

$$\mathcal{M}(\alpha) \approx \sum_{k\in[K]} w_k(\alpha) \sum_{i:\ \xi_i(\alpha)\in\Delta(k)} m_i e^{j\delta_k} = \sum_{k\in[K]} w_k(\alpha)\ z_k\ e^{j\delta_k}, \tag{26}$$

where the z_k is the sum of the magnitudes of data points in $\mathcal{I}_k$.[3] Next, let us assume that the data points are symmetrically distributed around a main direction. Then, for the values that $w_k(\alpha)$ is non-zero, the phases of symmetric samples' sets elide each other and we can approximate the summation with the absolute value, i.e. the following equation stands:

$$M(\alpha) = |\mathcal{M}(\alpha)| \approx \sum_{k\in[K]} w_k(\alpha)\ z_k = \mathbf{w}(\alpha)^\top \mathbf{z}. \tag{27}$$

Fig. 3 (right) shows the original points (green circles) for the same data matrix as in Fig. 1. In the same plot, the red circles represent the entries of $\mathbf{z}$. Fig. 4 plots the entries of $\mathbf{z}$ and $\mathbf{w}(\alpha)$ versus index $k \in [K]$ and offers an important insight: as α varies, the window defined by $\mathbf{w}(\alpha)$ periodically shifts over the lag k, covering different entries of $\mathbf{z}$. Similar to Colonnese et al. [8], we notice that for α taking values in $\{(k-1)2\pi/K\}_{k\in[K]}$, $S(\alpha)$ takes the values of the periodical convolution of the entries of $\mathbf{w} = \mathbf{w}(0)$ and $\mathbf{z}$.

A few remarks are in order. Let us notice that a small asymmetry in the data distribution around the principal component would affect the estimate of α_{opt}. This notwithstanding, its effect on the estimate of θopt would be restrained to the inclusion or exclusion of few samples at the extrema of the selected half-plane (see Fig. 4). Therefore, the estimate θopt is inherently robust with respect to the data distribution symmetry. Finally, let us observe that a different window (e.g. Hamming or Hanning) could also be adopted to deal with possible asymmetries of the data distribution, especially far from the main data direction or for small number of samples. The study of the impact of different windows is left for further study.

Based on the above, the search for the optimal half-plane α_{opt} in (17) can be rephrased as a cyclic shift parameter estimation problem. That is, we estimate α_{opt} in (17) as

$$\hat{\alpha} = \frac{2\pi}{K}(\hat{\nu} - 1), \tag{28}$$

where $\hat{\nu} = \text{argmax}_{l\in[K]}|g_l|$, $\mathbf{g} = \text{IDFT}(\tilde{\mathbf{w}} \odot \tilde{\mathbf{z}})$, $\tilde{\mathbf{w}} = \text{DFT}(\mathbf{w})$, and $\tilde{\mathbf{z}} = \text{DFT}(\mathbf{z})$. Operator $\odot$ represents the element-wise product of two vectors of the same size. The discrete-time Fourier transform (DFT) and inverse DFT (IDFT) are implemented by means of Fast Fourier Transform (FFT), with reduced cost $\mathcal{O}(K \log K)$. Finally, our algorithm returns $\mathbf{u}(\Phi(\hat{\alpha}))$ as an estimate of $\mathbf{q}_{L1}$.

That is, L1-PC estimation is carried out in two steps. First, we estimate the angle α_{opt} of the half-plane that contains L1-PC, by search in the quantized discrete set $\{(k-1)2\pi/K\}_{k\in[K]}$. Second, the L1-PC is estimated as $\mathbf{u}(\Phi(\hat{\alpha}))$. We notice that any errors in the estimation of α_{opt} result into rotations of the half plane on which the sum in (15) is computed and affect the final accuracy of the L1-PC estimation only if they cause erroneous inclusion/exclusion of data points at the borders of the half-plane.

Thereby, the fact that the final estimate θ belongs to a discrete Θ turns into an improved robustness of the L1 PCA estimator w.r.t. estimation errors on α.

3.3. Generalization to $D \geq 2$

The above analysis is straightforwardly generalized to the case $D \geq 2$.

To the aim of generalization, let us consider the polar representation of the ith observation vector in $\mathbb{R}^D$, $\mathbf{x}_i = m_i \mathbf{u}(\boldsymbol{\phi}_i)$, where $m_i = \|\mathbf{x}_i\|_2$, $\boldsymbol{\phi}_i \in [0, 2\pi)^{D-1}$, and $\mathbf{u}(\boldsymbol{\phi}_i)$ is a hyperspherical coordinate vector [9].[4] In a nutshell, the proposed method in the general case relies on the computation steps identified for the case $D = 2$, namely: (i) accumulation of the module of the samples as a function of the $D-1$ phases, (ii) computation of the $D-1$-dimensional DFT of the accumulation function, (iii) product with the DFT of the $D-1$-dimensional windowing function, and iv) IDFT of the product and identification of the maximum of the so obtained $D-1$-dimensional sequence.

In more detail, for $D \geq 2$ and $\boldsymbol{\alpha} \in [0, 2\pi)^{D-1}$, window vector $\mathbf{w}(\alpha)$ extends to an order-$(D-1)$ tensor $\mathcal{W}(\boldsymbol{\alpha}) \in \{0,1\}^{K\times\ldots\times K}$, defined in (29).

$$[\mathcal{W}(\boldsymbol{\alpha})]_{k_1,\ldots,k_{D-1}} = \begin{cases} 1, & \alpha_1 - \pi/2 < \frac{(k_1+1/2).2\pi}{K} \leq \alpha_1 + \pi/2, \ldots \\ & \alpha_{D-1} - \pi/2 < \frac{(k_{D-1}+1/2)2\pi}{K} \leq \alpha_{D-1} + \pi/2. \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

[3] We notice that $\mathbf{z}$ can be straightforwardly computed by an extended set of $2N$ measurements, including the original ones and their π-shifted versions.

[4] For $\boldsymbol{\phi} \in [0, 2\pi)^{D-1}$, we define $\mathbf{u}(\boldsymbol{\phi}) = [\cos(\phi_1), \sin(\phi_1)cos(\phi_2), \ldots, \sin(\phi_1)\ldots\sin(\phi_{D-2})\cos(\phi_{D-1}), \sin(\phi_1)\ldots\sin(\phi_{D-1})]^\top$.

Similarly, $\mathbf{z}$ extends to the order-$(D-1)$ tensor $\mathcal{Z} \in \mathbb{C}^{K\times\ldots\times K}$, where $[\mathcal{Z}]_{k_1,\ldots,k_{D-1}} = \sum_{i\in\mathcal{I}_{k_1,\ldots,k_{D-1}}} m_i$ and

$$\mathcal{I}_{k_1,\ldots,k_{D-1}} = \{i \in [N] : \forall d \ [\phi_i]_d \in \Delta(k_d) \text{ or } [\phi_i]_d + \pi \in \Delta(k_d)\}. \quad (30)$$

Then, the L1-PC in $\mathbb{R}^D$ is identified by first estimating

$$\hat{\mathbf{v}} = \underset{\mathbf{v}\in[K]^{D-1}}{\operatorname{argmax}} \left|[\mathcal{G}]_{v_1,\ldots,v_{D-1}}\right|, \quad (31)$$

where $\mathcal{G} = \text{IDFT}_{D-1}(\tilde{\mathcal{W}} \odot \tilde{\mathcal{Z}}), \tilde{\mathcal{W}} = \text{DFT}_{D-1}(\mathcal{W}(\mathbf{0}_{D-1}))$, and $\tilde{\mathcal{Z}} = \text{DFT}_{D-1}(\mathcal{Z})$. High-order DFT/IDFT are conducted by multiplying $\mathcal{Z}$ in all $D-1$ modes with the FFT matrix.

Then, we find

$$\hat{\boldsymbol{\alpha}} = \frac{2\pi}{K}(\hat{\mathbf{v}} - \mathbf{1}_{D-1}) \quad (32)$$

and return the L1-PC estimate as $\mathbf{u}(\Phi(\hat{\boldsymbol{\alpha}}))$, where

$$\Phi(\hat{\boldsymbol{\alpha}}) = [\arg(\mathbf{m}(\hat{\alpha}_1)), \ldots, \arg(\mathbf{m}(\hat{\alpha}_{D-1}))]^\top,$$

$$\mathbf{m}(\theta) = \sum_{i\in[N]} \|\mathbf{x}_i\|_2 \mathbf{u}(\boldsymbol{\phi}_i + \beta(\boldsymbol{\phi}_i, \theta)),$$

and

$$\beta(\boldsymbol{\phi}_i, \theta) = [\beta([\boldsymbol{\phi}_i]_1, \theta), \ldots, \beta([\boldsymbol{\phi}_i]_{D-1}, \theta)]^\top.$$

A summary of the proposed method is presented in Algorithm 1.

Algorithm 1 Proposed FFT-based L1-PC computation.

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{D\times N} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, where $\mathbf{x}_i = m_i \mathbf{u}(\boldsymbol{\phi}_i)$

s1. Compute $\tilde{\mathcal{W}} \leftarrow \text{DFT}(\mathbf{W}(\mathbf{0}))$.
s2. Build $\mathcal{Z}$ and Compute $\tilde{\mathcal{Z}} \leftarrow \text{DFT}(\mathbf{Z})$.
s3. Compute $\mathcal{G} \leftarrow \text{IDFT}(\tilde{\mathcal{W}} \odot \tilde{\mathcal{Z}})$.
s4. Find the index vector $\hat{\mathbf{v}}$ of the largest absolute entry in tensor $\mathcal{G}$.
s5. Set $\hat{\boldsymbol{\alpha}} = (\hat{\mathbf{v}} - \mathbf{1}_{D-1})2\pi/K$ and return $\mathbf{u}(\Phi(\hat{\boldsymbol{\alpha}}))$

3.4. Incremental algorithm computation

So far, we have presented a version of the algorithm working on a batch of data. An online, incremental computation of the best angle version is herein derived. Specifically, we observe that the formulation of the maximization in Eq. (31) allows to highlight the contribution of each and every incoming sample. Let as assume that an L1-PCA estimate is available given the set of available samples. Each new received sample requires just updating $\tilde{\mathcal{Z}}$ and computing the new maximum. For the sake of simplicity, we outline the incremental computation for the case $D = 2$ in more detail. Let us denote by $\mathbf{g}^{(n)} = \text{IDFT}(\tilde{\mathbf{w}} \odot \tilde{\mathbf{z}}^{(n)})$ the convolution computed up to the nth observation. When a new observation with absolute value m_{n+1} arrives, we have

$$\mathbf{g}^{(n+1)} = \text{IDFT}(\tilde{\mathbf{w}} \odot \tilde{\mathbf{z}}^{(n+1)}) = \text{IDFT}(\tilde{\mathbf{w}} \odot \tilde{\mathbf{z}}^{(n)}) + \text{IDFT}(\tilde{\mathbf{w}} \odot \Delta\tilde{\mathbf{z}}^{(n+1)})$$

being $\Delta\tilde{\mathbf{z}}^{(n+1)}$ the DFT of the incremental component. Let us denote by k_+, k_- the indexes of the phase intervals where the observation and its π-shifted version belong. With these positions, $\Delta\tilde{\mathbf{z}}^{(n+1)}$ is straightforwardly computed as follows: $\Delta\tilde{\mathbf{z}}^{(n+1)} = m_{n+1}\mathbf{e}(k_+) + m_{n+1}\mathbf{e}(k_-)$, being $\mathbf{e}(k)$ the DFT basis vector $\mathbf{e}(k) = [1 \ e^{-j\frac{2\pi}{K}k} \ldots e^{-j\frac{2\pi}{K}k(K-1)}]^T$. The element-wise DFT domain multiplication $\tilde{\mathbf{w}} \odot \Delta\tilde{\mathbf{z}}^{(n+1)}$ is then computed as follows: $\tilde{\mathbf{w}} \odot \Delta\tilde{\mathbf{z}}^{(n+1)} = m_{n+1}\tilde{\mathbf{w}} \odot \mathbf{e}(k_+) + m_{n+1}\tilde{\mathbf{w}} \odot \mathbf{e}(k_-)$. We recognize that the multiplication by $\mathbf{e}(k)$ in the DFT domain corresponds to a circular shift by k in the original domain. Thereby, denoting by $CS(\cdot, k)$ a vector cyclic shift operator by k, we straightforwardly compute $\mathbf{g}^{(n+1)}$ as

$$\mathbf{g}^{(n+1)} = \mathbf{g}^{(n)} + m_{n+1}\text{CS}(\mathbf{W}, k_+) + m_{n+1}\text{CS}(\mathbf{W}, k_-)$$

This operation requires K sums every new sample to compute the updated values $\mathbf{g}^{(n+1)}$; then, the maximum is updated by K comparisons. The above described procedure is straightforwardly extended to the case $D > 2$.

3.5. Remarks

The role of K K represents the number of angular windows and the FFT size. Therefore, it affects both computation cost and the precision with which the half-plane is determined. If K is too large with respect to the number of available samples N, z_k might not be estimated accurately, affected by an erratic error known as population noise. On the other hand, when both N and K tend to infinity, the entries of $\mathbf{z}$ tend to the magnitude weighted tomographic projection of the probability density function of the observed samples and the estimator of the cyclic shift α_{opt} is asymptotically efficient. In practice, both N and K are limited and K affects the quantization step with which the phase α_{opt} is estimated. Still, K affects the actual L1-PCA estimate only indirectly, namely through the summation in (25). This inherent robustness of the rank-1 component estimate assures that quantization error in estimating the half-plane direction may be accepted in order to assure a better estimate $\mathbf{z}$, which in turn directly affects the sum forming L1-PC estimate.

Complexity Finally, the computational complexity of the FFT based algorithm,[5] accounting the main steps of sign changing, FFT computing and max extraction on the FFT samples is approximated as

$$\mathcal{C}_{FFT} = O\big(N \cdot D + D \cdot K^D \log(K) + K\big)$$

(for comparison sake let us recall here that the computational complexity of the bit flipping algorithm [3] is $\mathcal{C}_{BF} = O\big(N \cdot D \cdot \min\{N, D\} + 3 \cdot N^2 + N \cdot D\big)$). Clearly, the complexity of the proposed method is dominated by the FFT computational cost, which increases with the data dimension D, whereas it smoothly scales with the number of measurements. Therefore, the method is well suited to the big data framework where, despite single measurements may have a restrained dimension, a huge number N of measurements needs to be processed. This makes the proposed FFT method suitable for certain big data applications, ranging from extended reality to bioinformatics, which process huge amounts of volumetric data, video point clouds, and more in general 2D Riemannian manifolds embedded in 3D spaces. In these cases, the proposed method provides a computationally efficient solution to the L1 PCA problem in (1) even for large N.

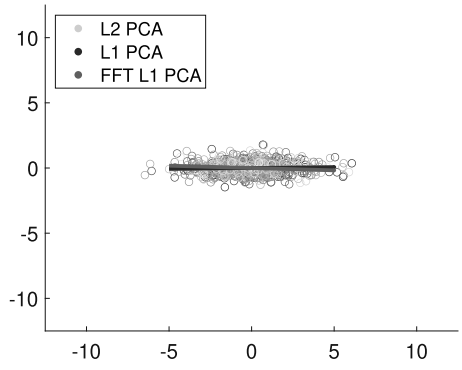
4. Numerical studies

Herein, we assess the performance of the proposed L1-PC estimator, also in the presence of outliers, comparing it with standard PCA and L1-PCA [3] alternatives.

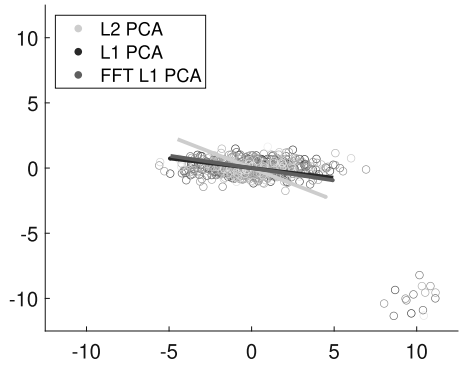
4.1. Synthetic data

We first provide examples in the case $D = 2$. In Fig. 5 we show an example of L1-PC estimated on a data matrix $\mathbf{X}$, drawn from multivariate Gaussian distribution ($N = 625$, $D = 3$). The proposed

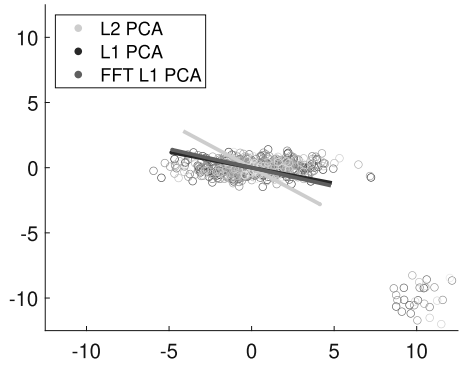
[5] We report here the complexity of the batch algorithm; let us observe that after the computation of the L1-PC over the first batch the computational complexity of the incremental version of the algorithm is lower and still linear with number of measurements N.



(a) $p = 0\%$



(b) $p = 2.5\%$



(c) $p = 5\%$

Fig. 5. Example of rank-1 L1 PCA estimated on a multivariate normal sample data set **X** for different percentages of outliers: (a) $p = 0$; (b) $p = 2.5\%$ and (c) $p = 5\%$.

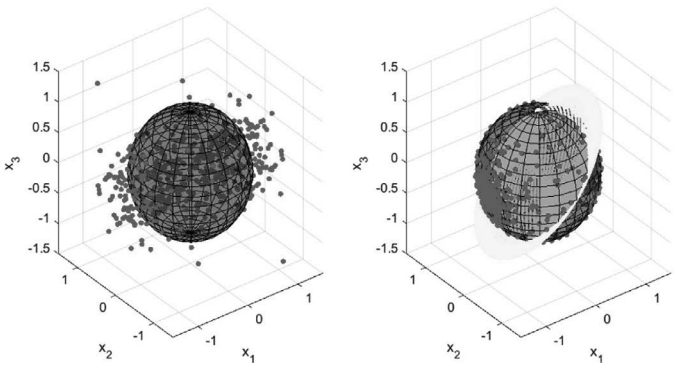


Fig. 6. Data matrix **X** drawn from multivariate Gaussian distribution ($N = 625$, $D = 3$): original measurements (left plot) and projection on the unit sphere and plane separating the two half-spaces (right plot).

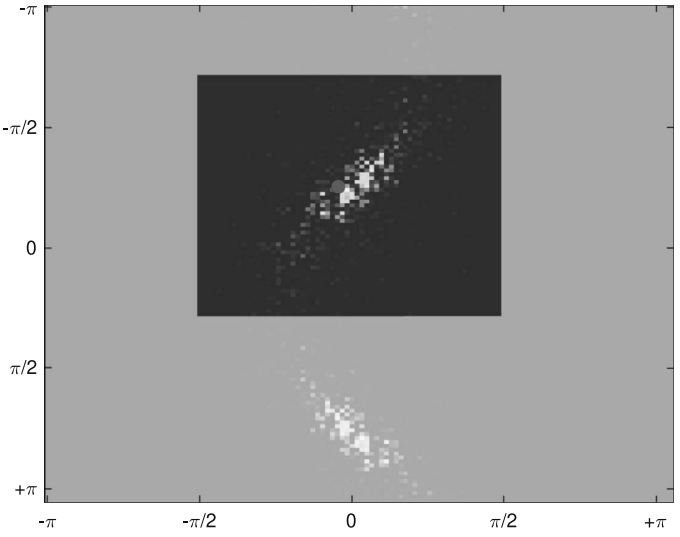


Fig. 7. Estimated entries of **z** and window **w** ($N = 625$, $D = 3$).

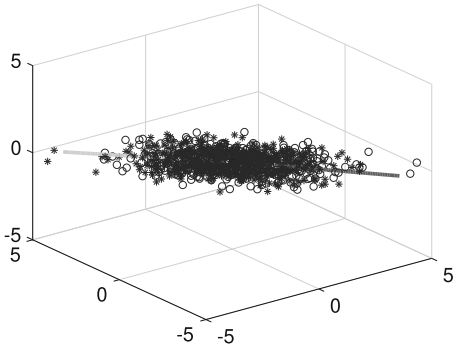


Fig. 8. Estimated L1-PC using the algorithm in Markopoulos et al. [3] and the proposed fast FFT based method.

fast FFT-based estimator maintains the accuracy and the resilience property of the L1-PCA estimator.

Furthermore, we offer an example for $D = 3$, with a data matrix **X** drawn from multivariate Gaussian distribution. The data and their projection on the unit sphere are shown in Fig. 6 ($N = 625$). To elaborate, we consider the bi-dimensional interval $(\phi_1, \phi_2) \in (-\pi, \pi) \times (-\pi, \pi)$ and we quantize it into $K \times K$ intervals of width $2\pi/K \times 2\pi/K$. The magnitudes of the received samples are accumulated in **z**. Besides, a bi-dimensional window function $[\mathcal{W}([\theta_1, \theta_2]^\top)]_{k_1,k_2}$ equal to 1 for $\theta_1 - \pi/2 < \frac{(k_1+1/2)2\pi}{K} \leq \theta_1 + \pi/2, \theta_2 - \pi/2 < \frac{(k_2+1/2)2\pi}{K} \leq \theta_2 + \pi/2$ and zero otherwise is considered. After computing the circular convolution of the above defined functions, the optimal window displacement is found, as shown in Fig. 7 The orientation of the optimal L1-PC is determined by the optimal azimuth and elevation $\arg\max_{k_1,k_2} |[\text{IDFT}_2(\tilde{\mathcal{W}} \odot \tilde{\mathcal{Z}})]_{k_1,k_2}|$. The estimated L1-PC, obtained using the algorithm in [3] and the proposed fast FFT based method are shown in Fig. 8.

The accuracy of the proposed FFT-based L1-PC is similar to that of the algorithm in Markopoulos et al. [3], as it can be observed in

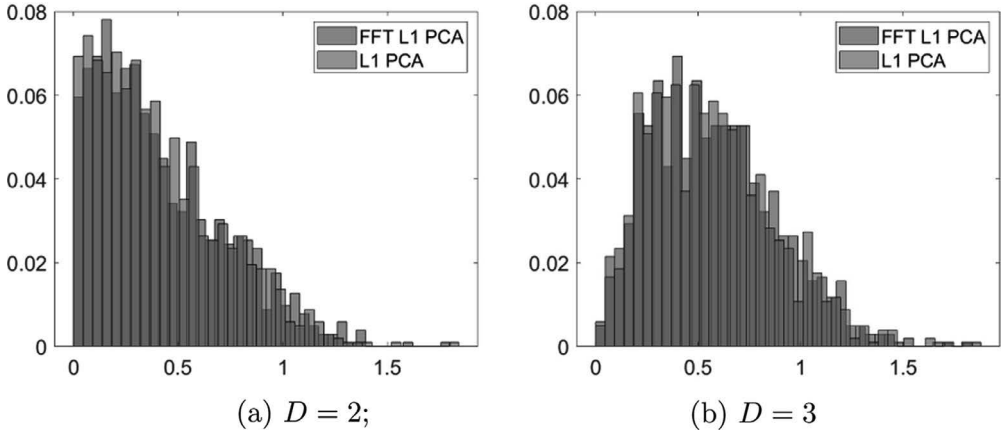


Fig. 9. Histograms of the quadratic reconstruction error $||\mathbf{x} - \mathbf{q}\,\mathbf{q}^\top\mathbf{x}||_2$ on a multivariate Gaussian distribution ($N = 512, K = 128$).

Fig. 9, where the histogram of the quadratic reconstruction error $||\mathbf{x} - \mathbf{q}\,\mathbf{q}^\top\mathbf{x}||_2$ observed on the N samples is computed.

We now compare the proposed approach with i) the approximated the L1-PC obtained by substituting b_{opt} with the ± 1 quantization of the highest eigenvalue eigenvector of the $N \times N$ matrix $\mathbf{X}^\top\mathbf{X}$ (denoted as EIG in the following) ii) the fixed-point iteration method of [10] (denoted as FP in the following). In Fig. 10 we present the Mean Square Error obtained by the proposed FFT algorithm (FFT), the algorithm in [3] (L1), the L2-PCA (L2), and the two above mentioned competitors EIG and FP, for the case $D = 2$ (left) $D = 3$ (right) ($N = 2048$, $p = 5\%$, $K = 128$, 20 Montecarlo runs). The proposed fast FFT based algorithm maintains the resilience property of classical algorithms, and it proves more resilient than fast state-of-the-art competitors.

In Fig. 11, we plot the Mean Square Error (MSE) per run versus the algorithm Computational complexity for the proposed FFT algorithm (black points) and the fast competitors EIG (cyan points), and FP [10] (blue points) for $D = 2$ (left) and $D = 3$ (right) ($N = 1024$, $p = 5\%$, $K = 128$, 20 Montecarlo runs).[6] We appreciate the reduced MSE achieved by the FFT based, at a computational complexity comparable to the fastest but less accurate algorithm in Kwak [10]. Let us remark that this favourable condition is observed for small D, and large N. This condition is found in different applications dealing with big data acquired on a plane or a surface. A relevant example is represented by point clouds, which are leveraged in emerging processing fields such as extended/augmented reality applications.

As far as point clouds are concerned, a relevant application of the proposed method is that of point cloud registration, which exploits L1-PC computation. Due to noisy acquisition and huge number of measurements, conventional methods are error prone or computationally heavy. In Fig. 12 a 3D point cloud of $N = 35,947$, points, each of dimension $D = 3$, is plotted, for no outliers (left) and 5% outliers (right). For both the clouds the rank-1 L1 PCA is computed using the FFT-based method ($K = 128$). The estimate appears stable with respect to the noise.

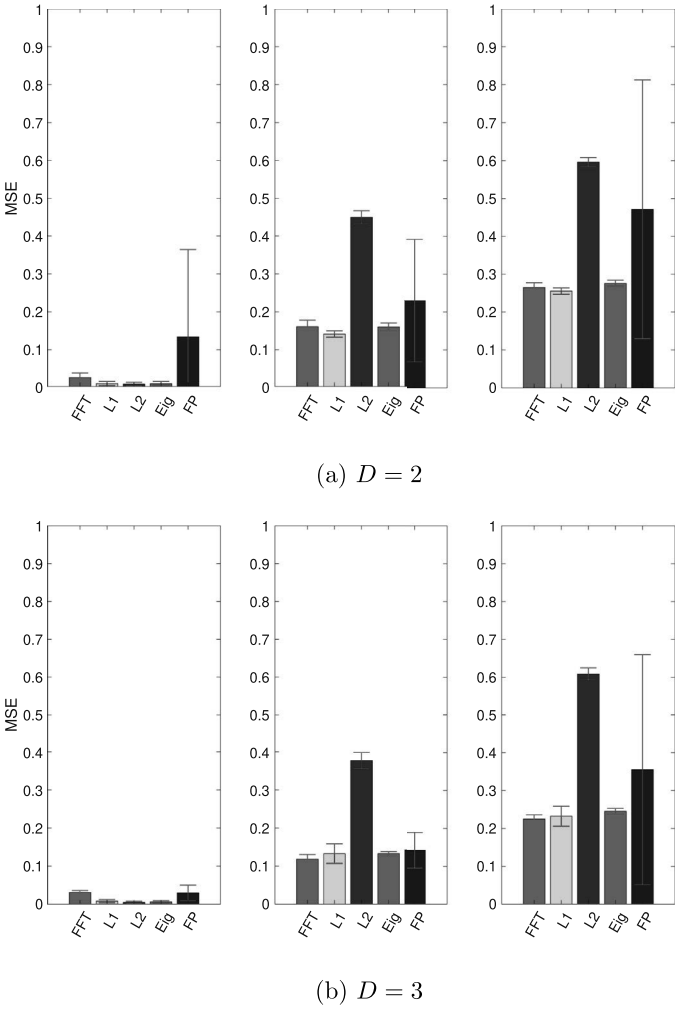


Fig. 10. MSE obtained by the proposed FFT algorithm, the algorithm in Markopoulos et al. [3], the L2-PCA and the three above mentioned competitors, for the case $D = 2$ (left) $D = 3$ (right) ($N = 1024$, $p = 5\%$, $K = 128$, 20 Montecarlo runs).

4.2. Real data

We now provide results of applications of the FFT based L1 PCA method to real data. Specifically, we selected two point clouds retrieved from 3D scanning of real objects' dataset as described in Li et al. [11].[7] Each point cloud represents a real 3D object (Ikea chair) by a number of 3D points N equal to 9400 and 26,001, respectively. For both the point clouds, we have computed the

[6] For the sake of simplicity, the EIG algorithm computational complexity has been approximated as $CC_{EIG} \approx N^2$, whereas for the algorithm in Kwak [10] we set $CC_{FP} \approx 4NDN_{iter}$, $N_{iter} = 10$

[7] The dataset is available inline at http://graphics.stanford.edu/projects/objectsensing/

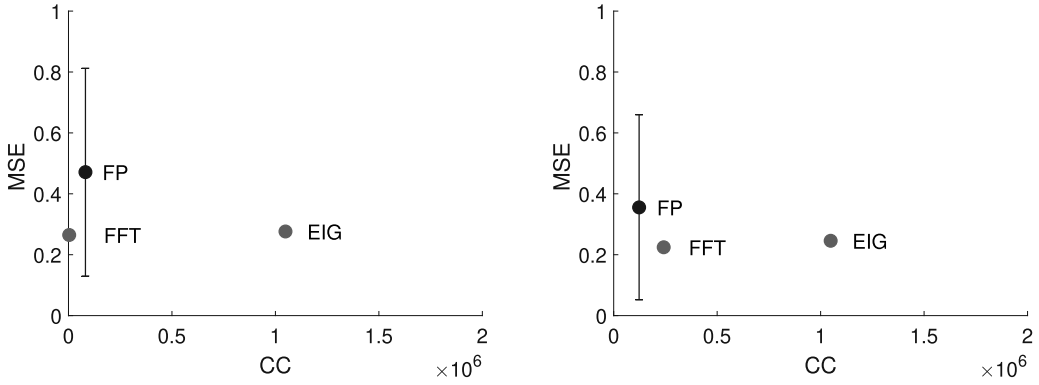


Fig. 11. MSE versus computational complexity (CC) by the proposed FFT algorithm (red points), the EIG algorithm (gray points), and the algorithm in Kwak [10] (blue points), for the cases $D = 2$ (left) and $D = 3$ (right) ($N = 1024$, $p = 5\%$, $K = 128$, 20 Montecarlo runs). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

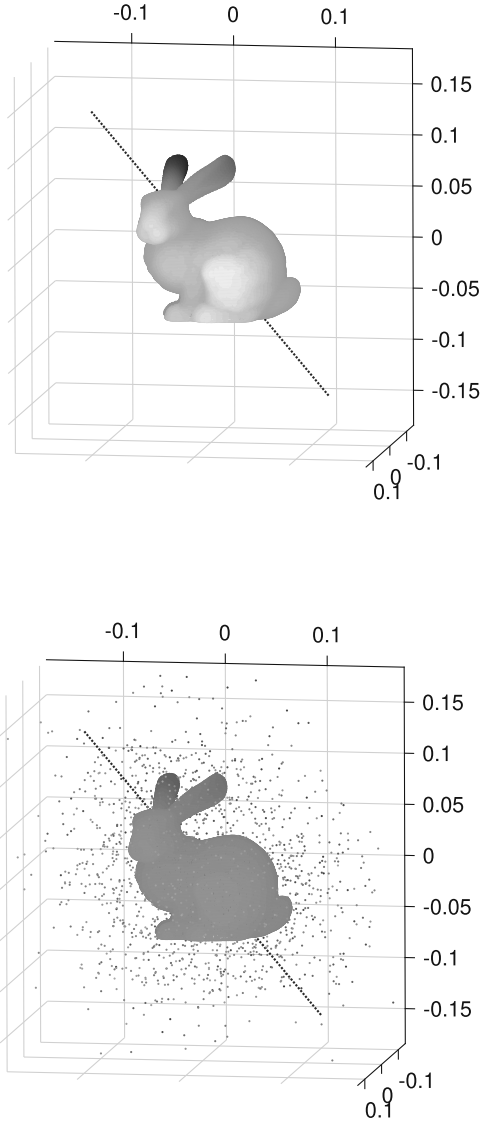


Fig. 12. Point cloud dataset "Stanford Bunny" (left); Noisy point cloud dataset and L1-PC (right), for $N = 35947$, $D = 3$.

rank-1 L1-PCA on the original data and on a version obtained by adding a white zero-mean Gaussian noise of standard deviation equal to $\sigma_n = 0.1$. The results are shown in the Figs. 13 and 14, where the point clouds (colored points) and the rank-1 L1-PCA direction (red line) are shown. We recognize that the

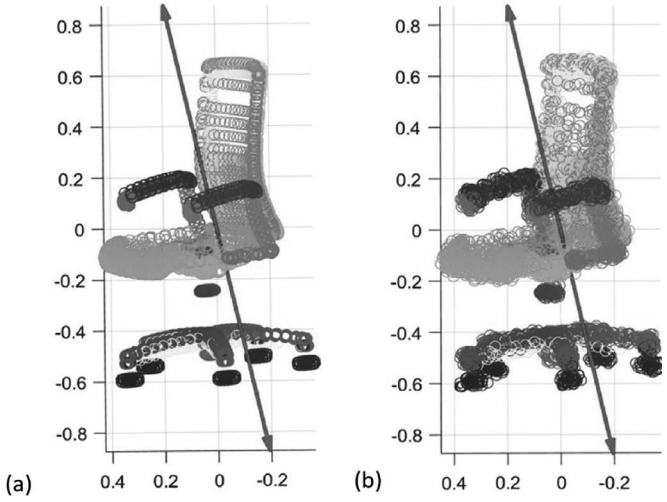


Fig. 13. Point cloud data retrieved via real object scanning: original point cloud (a), noisy version (b), rank-1 L1-PCAs (red lines) ($N = 9400$, $K = 64$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

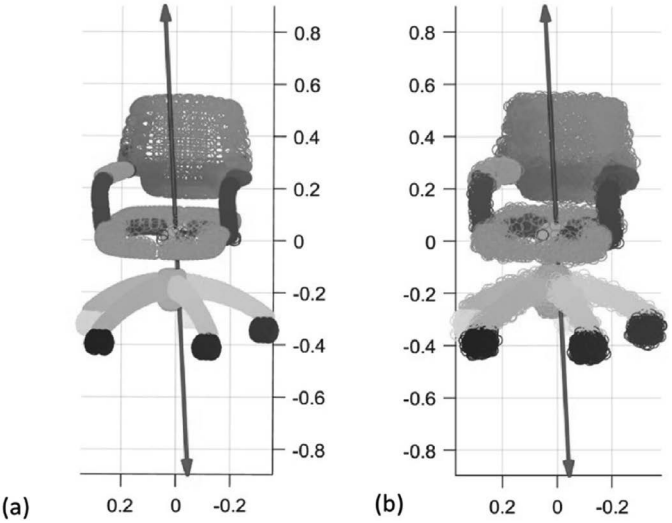


Fig. 14. Point cloud data retrieved via real object scanning: original point cloud (a), noisy version (b), rank-1 L1-PCAs (red lines) ($N = 26001$, $K = 64$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

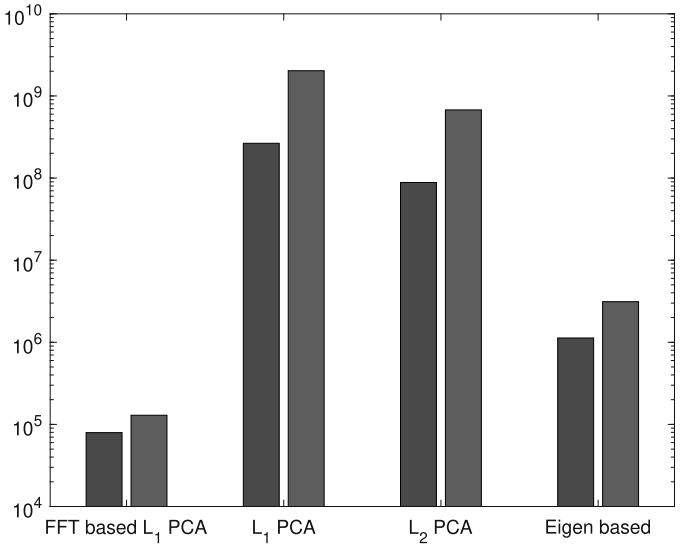


Fig. 15. Computational complexity (log-scale) of rank-1 L1-PCA estimation on the point cloud data retrieved via real object scanning using $K = 64$, with $N = 9400$ (blue bars) and $N = 26001$ (red bars). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

rank-1 L1-PCA direction captures the main axis of the object and it is stable w.r.t. the additive noise. Besides, we report the computational complexity of FFT-based rank-1 L1-PCA on the two point clouds in Fig. 15. For the sake of comparison, we report the complexity of state-of-the-art algorithms for the same parameters. We recognize that the FFT-based algorithm is computationally efficient and it is a promising analysis tool for volumetric data analysis.

5. Conclusions

We reformulated L1-PCA into a cyclic shift parameter estimation problem and presented a fast estimator for the L1-PC based on FFT. Compared to counterparts, our method is accurate and computationally efficient. The choice of the FFT size allows for a trade-off between computational complexity and accuracy. Numerical simulations assess the algorithm's performance in a variety of experimental conditions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Stefania Colonnese: Conceptualization, Data curation, Writing – original draft. **Panos P. Markopoulos:** Formal analysis, Resources, Writing – original draft. **Gaetano Scarano:** Methodology, Writing – review & editing. **Dimitris A. Pados:** Methodology, Supervision, Writing – review & editing.

Acknowledgments

The work of P. P. Markopoulos was also supported in part by the National Science Foundation, under grant OAC-1808582, and by the Air Force Office of Scientific Research (Young Investigator Program) under grant FA9550-20-1-0039.

References

[1] R. Martín-Clemente, V. Zarzoso, Unsupervised classification of array data based on the L1-norm, in: 2018 52nd Asilomar Conference on Signals, Systems, and Computers, IEEE, 2018. 343–176
[2] R. Martin-Clemente, V. Zarzoso, LDA via L1-PCA of whitened data, IEEE Trans. Signal Process. 68 (2019) 225–240.
[3] P.P. Markopoulos, S. Kundu, S. Chamadia, D.A. Pados, Efficient l1-norm principal-component analysis via bit flipping, IEEE Trans. Signal Process. 65 (16) (2017) 4252–4264.
[4] C. Kim, D. Klabjan, A simple and fast algorithm for l1-norm kernel PCA, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2019) 1842–1855.
[5] S. Hauberg, A. Feragen, M.J. Black, Grassmann averages for scalable robust PCA, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3810–3817.
[6] X. Song, An intuitive and most efficient Ll-norm principal component analysis algorithm for big data, in: 2019 53rd Annual Conference on Information Sciences and Systems (CISS), IEEE, 2019, pp. 1–4.
[7] B. Minnehan, A. Savakis, Grassmann manifold optimization for fast L_1-norm principal component analysis, IEEE Signal Process. Lett. 26 (2) (2018) 242–246.
[8] S. Colonnese, S. Rinauro, G. Scarano, Generalized method of moments estimation of location parameters: application to blind phase acquisition, IEEE Trans. Signal Process. 58 (9) (2010) 4735–4749.
[9] M. Al-Khassaweneh, M. Villafane-Delgado, A.Y. Mutlu, S. Aviyente, A measure of multivariate phase synchrony using hyperdimensional geometry, IEEE Trans. Signal Process. 64 (11) (2016) 2774–2787.
[10] N. Kwak, Principal component analysis based on L1-norm maximization, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1672–1680.
[11] Y. Li, A. Dai, L. Guibas, M. Nießner, Database-assisted object retrieval for real–time 3Dreconstruction, in: Computer Graphics Forum, 34, Wiley Online Library, 2015, pp. 435–446.