Abstract

We present an efficient algorithm for standard ICA that needs only a nearly linear number of samples and has polynomial time complexity. The algorithm is a recursive version of the Fourier PCA method of Goyal et al. [12]. Its analysis is based on properties of random polynomials, namely the spacings of an ensemble of polynomials.
1 Introduction

Recovering latent structure from unlabeled data is the broad area of unsupervised learning. A large number of problems in this area are based on generative models of data, typically probability distributions defined by a finite set of parameters. Two well-known instances are the problems of learning a mixture of Gaussians and finding planted cliques in random graphs: in both cases, the goal is to find clusters in the data which group “similar” or “close” items together. As data abounds, and labels are expensive, efficient methods for unsupervised learning are gaining importance beyond their mathematical elegance.

Here we consider the classic problem of Independent Component Analysis (ICA), which originated in signal processing and has become a fundamental problem in machine learning and statistics, finding applications in diverse areas, including neuroscience, computer vision and telecommunications. The input to the problem is a set of i.i.d. vectors from a distribution in $\mathbb{R}^n$. The latter is assumed to be an unknown linear transformation of an unknown distribution with independent 1-dimensional component distributions. More precisely, each observation $x \in \mathbb{R}^n$ can be written as $x = As$, where $A \in \mathbb{R}^{n \times n}$ is an unknown matrix and $s \in \mathbb{R}^n$ has components $s_1, \ldots, s_n \in \mathbb{R}$ generated independently (from possibly different one-dimensional distributions). ICA is the problem of estimating the matrix $A$, the basis of the latent product distribution, up to a scaling of each column and a desired error $\epsilon$. One cannot hope to recover $A$ if more than one $s_i$ is Gaussian — any set of orthogonal directions in the subspace spanned by Gaussian components would also be consistent with the model. Hence the model also assumes that at most one component is Gaussian and the other component distributions differ from being Gaussian in some fashion; the most common assumption is that the fourth cumulant, also called the kurtosis, is nonzero (it is zero for a Gaussian).

Our main result is a polynomial-time algorithm for ICA, under the fourth cumulant assumption, using only $\tilde{O}(n)$ samples, which is nearly optimal. This improves on previous polynomial-time algorithms, which all require a higher polynomial number of samples ($\Omega(n^5)$ or higher). It applies to learning mixtures of spherical Gaussians and can be extended to the setting where the model is corrupted with Gaussian noise. Before stating our main result precisely, we place it in the context of related work.

1.1 Related work

The literature on ICA, the most well-known model for blind source separation, is vast, with many proposed algorithms (see [6, 17] for an excellent survey of the older literature). ICA is an important statistical/machine-learning model [14], and has found applications in many data driven fields. Recently, it has received fresh interest as a tool for sparsifying layers in deep neural nets [18]. Frieze et al. [11] were the first to provide rigorous finite sample guarantees, with several recent papers improving their guarantee for the fully determined case when $A$ is nonsingular [19, 3, 4, 2]. These results either assume that component distributions are specific or that the fourth moment is bounded away from that of a Gaussian.

Goyal et al. [12] recently gave an algorithm called Fourier PCA that can deal with differences from being Gaussian at any moment. They use cumulants as a notion of difference from being a Gaussian. The $r$’th cumulant of random variable $x$, denoted by $\text{cum}_r(x)$, is a polynomial of the first $r$ moments of $x$; the first two cumulants are the mean and the variance and the fourth is $\mathbb{E}(x^4) - 3\mathbb{E}(x^2)^2$. A Gaussian has all but the first two cumulants equal to zero, and any distribution that has only a finite number of nonzero cumulants must be Gaussian. We now state their result
for the fully determined setting of nonsingular $A$.

**Theorem 1.1.** Let $x \in \mathbb{R}^n$ be given by an ICA model $x = As$, where $A \in \mathbb{R}^{n \times n}$ columns of $A$ have unit norm and let $\sigma_n(A) > 0$. Suppose that for each $s_i$, there exists a $k_i \leq k$ such that $|\text{cum}_{k_i}(s_i)| \geq \Delta > 0$ and $\mathbb{E}(|s_i|^k) \leq M$. Then, one can recover the columns of $A$ up to signs, and to accuracy $\epsilon$ in polynomial time using $\text{poly}(n^k, M^k, 1/\Delta^k, 1/\sigma_n(A)^k, 1/\epsilon)$ samples with high probability.

The algorithm, called Fourier PCA, can handle unknown Gaussian noise. It extends to the underdetermined setting where the signal $s$ has more components than the observation $x$ (so $A$ is rectangular), resulting in a polynomial-time algorithm under a much milder condition than the nonsingularity of $A$.

The main technique is efficient tensor decomposition. For a Hermitian matrix $A \in \mathbb{R}^{n \times n}$, one can give an orthogonal decomposition into rank 1 components:

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T.$$

This decomposition, especially when applied to covariance matrices, is a powerful tool in machine learning and theoretical computer science. The generalization of this to tensors is not straight-forward, and many versions of this decomposition lead directly to NP-hard problems [15]. The application of tensor decomposition to ICA was proposed by Cardoso [5]. Such decompositions were used by Anandkumar et al. [1] and Hsu and Kakade [16] to give provable algorithms for various latent variable models. Goyal et al. extended these decompositions to a more general setting where the rank-one factors need not be linearly independent (and thus might be many more than the dimension).

In spite of these polynomial algorithms, even for standard ICA with a square matrix $A$, the dependence of the time and sample complexity on the conditioning of $A$ and the dimension $n$ make them impractical even in moderately high dimension. The construction of the 4th (or higher order) tensors for ICA has sample complexity that grows as a prohibitively high polynomial in $n$.

**1.2 Our result**

Our main result is a polynomial-time algorithm that uses a nearly linear number of samples. The assumption that $A$ is unitary can be achieved by first placing the input distribution in isotropic position.

**Theorem 1.2.** Let $x \in \mathbb{R}^n$ be given by an ICA model $x = As$, where $A \in \mathbb{R}^{n \times n}$ is unitary, the $s_i$ are independent with $\|s\| \leq K\sqrt{n}$ almost surely, and for each $i$, $|\text{cum}_4(s_i)| = \left| \mathbb{E}(|s_i|^4) - 3\mathbb{E}(|s_i|^2)^2 \right| \geq \Delta$. Let $M_5 = \max_j \mathbb{E}(|s_j|^5)$. Then, for any $\epsilon > 0$, Recursive FPCA with

$$\sigma = \frac{\Delta}{1000(\log^{5/2} n)M_5}$$

will, with high probability, recover vectors $\{b_1, \ldots, b_n\}$ such that there exist signs $\xi_i = \pm 1$ satisfying

$$\left\|A^{(i)} - \xi_i b_i \right\| \leq \epsilon$$

for each column $A^{(i)}$ of $A$, using $O(n \cdot K^2 M_5^2 \log^{13} n/\Delta^6 \epsilon^2) = O^*(n)$ samples.
The algorithm is a recursive variant of Fourier PCA. Roughly speaking, Fourier PCA proceeds by computing the eigenvectors of a very special (random) matrix associated with the data; these eigenvectors are essentially the columns $A^{(i)}$. To do so accurately, however, requires that the eigenvalues of the random matrix are well spaced. To accurately estimate eigenvector $v_i$, we need the corresponding eigenvalue $\lambda_i$ to be far from its immediate neighbors $\lambda_{i-1}$ and $\lambda_{i+1}$. To estimate all the eigenvectors, we need large spacings between all $n - 1$ adjacent eigenvalue pairs. In other words, we need that $\min_i \lambda_{i+1} - \lambda_i$ should be large, and the complexity of this method is polynomial in the inverse of the minimum gap.

Our new algorithm is less direct. Asking for good separation between all the eigenvalues is too much — instead of partitioning the eigenvectors into singleton sets, we simply group the eigenvectors according to which side of the largest gap $\max_i \lambda_{i+1} - \lambda_i$ they fall. The maximum gap is substantially larger than the minimum gap, and thus, the two subspaces we obtain — following the inverse proportionality argument above — are very accurate. On the other hand, the vectors inside either of these subspaces are not necessarily close to the desired $A^{(i)}$, but we can proceed recursively in each subspace, reusing the initial sample. The key fact though, is that at each stage, we only need the maximum gap to be large, and thus the number of samples is much smaller. As a motivating example, if we pick $n$ random points from $N(0, 1)$, the minimum gap is about $O(1/n^2)$ while the maximum gap in expectation is $\Omega(1/\sqrt{\log n})$. Since the sample complexity grows as the square of the inverse of this gap, this simple idea results in a huge saving.

The best tensor based algorithms all must construct the fourth moment tensor from samples (either explicitly or implicitly); this leads to an information-theoretic lower bound of $O(n^2)$ samples \[13\] and all previous algorithms \[11, 19, 3, 4, 2, 12\] use $\Omega(n^5)$ samples. Our algorithm needs only $O(n)$ samples. Its worst-case time complexity is bounded by at most $n$ singular value decompositions of real matrices of size $n \times n$ or smaller.

Our contributions are threefold: first, an ICA algorithm with nearly linear and thus nearly optimal sample complexity; second, the use and analysis of maximum spacings of the eigenvalues of random matrices as a tool for the design and analysis of algorithms (typically, one tries to control the minimum, and hence, all the gaps); and finally, our proof of the maximum spacing uses a simple but powerful coupling technique that allows for decoupling of rather complicated dependent processes. We note that our algorithmic result can be applied to learning mixtures of spherical Gaussians with linearly independent means and to ICA with Gaussian noise where $x = As + \eta$ and $\eta$ is from an unknown (not necessarily spherical) Gaussian distribution. We do not treat these extensions in detail here as they are similar to \[12\], but with the improved sample complexity of the core algorithm.

2 Outline of approach

For a random vector $x \in \mathbb{R}^n$ distributed according to $f$, the characteristic function is given by the Fourier transform

$$\phi(u) = \mathbb{E} \left( e^{iu^T x} \right) = \int f(x) e^{iu^T x} dx.$$ 

In our case, $x$ will be the observed data in the ICA problem. The second characteristic function or cumulant generating function given by $\psi(u) = \log(\phi(u))$. For $x = As$, we define the component-wise
characteristic functions with respect to the underlying signal variables

\[ \phi_j(u_j) = \mathbb{E}(e^{iu_j s_j}) \quad \text{and} \quad \psi_i(u_j) = \log(\phi_j(u_j)) = \sum_{k=1}^{\infty} \text{cum}_k(s_j)(iu_j)^k/k! . \] (1)

Here \( \text{cum}_k(y) \) is the \( k \)'th cumulant of the random variable \( y \), a polynomial in its first \( k \) moments (the second characteristic function is thus also called the cumulant generating function). Note that both these functions are with respect to the underlying random variables \( s_i \) and not the observed random variables \( x_i \). For convenience, we write \( g_i = \psi''_i \).

The reweighted covariance matrix in the algorithm is precisely the Hessian \( D^2 \psi \):

\[
D^2 \psi = -\frac{\mathbb{E}\left((x - \mu_u)(x - \mu_u)^T e^{iu^T x}\right)}{\mathbb{E}(e^{iu^T x})} = \Sigma_u,
\]

where \( \mu_u = \mathbb{E}\left(x e^{iu^T x}\right) / \mathbb{E}(e^{iu^T x}) \). This matrix \( D^2 \psi \) has a very special form; suppose that \( A = I_n \):

\[
\psi(u) = \log \left( \mathbb{E}(e^{iu^T s}) \right) = \log \left( \mathbb{E}\left( \prod_{j=1}^{n} e^{iu_j s_j} \right) \right) = \sum_{j=1}^{n} \log(\mathbb{E}(e^{iu_j s_j})) = \sum_{j=1}^{n} \psi_j(u_j).
\]

Taking a derivative will leave only a single term

\[
\frac{\partial \psi}{\partial u_j} = \psi'_j(u_j).
\]

And taking a second derivative will leave only the diagonal terms

\[
D^2 \psi = \text{diag} \left( \psi''_j(u_j) \right) = \text{diag} \left( g_j(u_j) \right).
\]

Thus, diagonalizing this matrix will give us the columns of \( A = I_n \), provided that the eigenvalues of \( D^2 \psi \) are nondegenerate. The general case for \( A \neq I_n \) follows from the chain rule. The matrix \( D^2 \psi \) is symmetric (with complex eigenvalues), but not Hermitian; it has the following decomposition as observed by Yeredor \[23\]. The statement below holds for any nonsingular matrix \( A \), we use it for unitary \( A \), since we can first place \( x \) in isotropic position so that \( A \) will be effectively unitary.

**Lemma 2.1.** Let \( x \in \mathbb{R}^n \) be given by an ICA model \( x = As \) where \( A \in \mathbb{R}^{n \times n} \) is nonsingular and \( s \in \mathbb{R}^n \) is an independent random vector. Then

\[
D^2 \psi = \text{Adiag} \left( g_i((A^T u)_i) \right) A^T.
\]

To obtain a robust algorithm, we rely on the eigenvalues of \( D^2 \psi \) being adequately spaced (so that the error arising from sampling does not mix the columns of \( A \)). To ensure this, we inject some randomness by picking a random Fourier coefficient, so that the \( g_i(u_i) \) are sufficiently anti-concentrated. We pick the random Fourier coefficient \( u \) according to a Gaussian \( N(0, \sigma^2 I_n) \). Goyal et al. \[12\] had to choose \( \sigma \) small enough and therefore the number of samples large enough so that with high probability for all pairs \( i \neq j \), they could guarantee

\[
|g_i((A^T u)_i) - g_j((A^T u)_j)| \geq \delta
\]
for a suitable $\delta$, leading to the polynomial complexity of Theorem 1.1.

Here we give an algorithm for the traditional case of fully-determined ICA, when all the latent variables $s_i$ differ from Gaussian in the fourth moment by at least $\Delta$. The algorithm we give is extremely efficient in terms of sample complexity. The major insight is that instead of having to space all the eigenvalues along the real line, it simply suffices if there is a single large gap between some adjacent pair of eigenvalues on the real line. As we show in Theorem 4.1 for quadratic polynomials evaluated over Gaussians, the maximum gap grows as $\Omega(1/\log n)$, while the minimum gap is $O(1/n^2)$. We can thus partition the eigenvectors into two sets according to where their eigenvalues fall relative to this large gap. We can then project the samples onto the two subspaces spanned by these sets and proceed recursively, reusing the initial sample.

3 Recursive FPCA

**Recursive Fourier PCA** ($\sigma$, Projection matrix $P \in \mathbb{R}^{n \times k}$)

1. (Termination check) If $k = 1$, return $P$.
2. (Projection) Project all samples by multiplying by $P^T$ to projected samples $S$.
3. (Isotropy) Find an isotropic transformation $B^{-1}$ with
   \[ B^2 = \frac{1}{|S|} \sum_{x \in S} (x - \bar{x})(x - \bar{x})^T. \]
4. (Fourier weights) Pick a random vector $u$ from $N(0, \sigma^2 I_k)$. For every $x$ in a new sample $S$, compute $y = B^{-1}x$, and its Fourier weight
   \[ w(y) = \frac{e^{iu^T y}}{\sum_{y \in S} e^{iu^T y}}. \]
5. (Reweighted Covariance) Compute the covariance matrix of the points $y$ reweighted by $w(y)$
   \[ \mu_u = \frac{1}{|S|} \sum_{y \in S} w(y)y \quad \text{and} \quad \Sigma_u = -\frac{1}{|S|} \sum_{y \in S} w(y)(y - \mu_u)(y - \mu_u)^T. \]
6. (SVD) Compute the spectral decomposition $\{\lambda_i\}, \{v_i\}$ of $\text{Re}(\Sigma_u)$.
7. (Eigenvalue gap) Find the largest gap $\lambda_{i+1} - \lambda_i$. If the gap is too small, pick $u$ again.
   Partition the eigenvectors into $V_1 = \{v_1, \ldots, v_i\}$ and $V_2 = \{v_{i+1}, \ldots, v_k\}$.
8. (Recursion) $W_1 = \text{Recursive FPCA}(\sigma, PV_1)$ and $W_2 = \text{Recursive FPCA}(\sigma, PV_2)$
9. Return $[W_1 \quad W_2]$. 
Analysis

The analysis of the recursive algorithm has three parts. We will show that:

1. There is a large gap in the set if diagonal values, i.e., the set \( \{ g_i((A^T u)_i) \} \). Since we make the distribution isotropic, \( A^T u \) has the same distribution as \( u \), so we can focus on \( g_i \) evaluated at independent Gaussians.

2. There is a partition of the columns of \( A \) into two subsets whose spans are \( V \) and \( \bar{V} \), so that the two subspaces obtained in the algorithm as the span of all eigenvectors above the largest gap and below this gap are close to \( V, \bar{V} \). This will follow using a version of Wedin’s theorem for perturbations of matrices.

3. The total error accumulated by recursion remains below the target error \( \epsilon \) for each column.

4.1 Maximum spacings of Gaussian polynomials

Here we study the largest gap between successive eigenvalues of the matrix \( D^2 \log(\phi(u)) \). For a set of real numbers \( x_1, \ldots, x_n \), define the maximum gap function as:

\[
\text{maxgap}(x_1, \ldots, x_n) = \max_{i \in [n]} \min_{j \in [n]: x_j \geq x_i} x_j - x_i
\]

The maxgap function is simply the largest gap between two successive elements in sorted order.

**Theorem 4.1.** Let \( \{ p_1(x), \ldots, p_n(x) \} \) be a set of \( n \) quadratic polynomials of the form \( p_i(x) = a_i x^2 \) where \( a_i > 0 \) for all \( i \) and \( \{ z_1, \ldots, z_n \} \) be iid standard Gaussians. Then, with probability at least \( \frac{1}{2000 \log^2 n} \),

\[
\text{maxgap}(p_1(z_1), \ldots, p_n(z_n)) \geq \frac{\log 2}{10 \log n} \min_i a_i.
\]

We can simply repeat the experiment \( O(\log^3 n) \) times to obtain a high probability guarantee. This type of maxgap function has been somewhat studied in the mathematics literature – there are a number of asymptotic results [9, 8, 10]. The rough intuition of these results is that asymptotically, the maxgap depends only on the tails of the random variables in question. Our work differs from these results in two very important ways – firstly, our results are quantitative (i.e., not simply in the limit of \( n \to \infty \)), and secondly, our result is true even if you pick the polynomial after fixing \( n \). The latter, in particular, makes the problem quite a bit harder as now the family of random variables is no longer even uniform over \( n! \).

**Proof.** The first stage of the proof is to reduce the problem from the random model \( \{ a_1 z_1^2, \ldots, a_n z_n^2 \} \) to sampling from a mixture model, which will more easily allow us to analyse the maximum gaps. To this end, let \( f_i \) denote the distribution of \( p_i(z_i) \), then consider the following uniform mixture model:

\[
F(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) = \frac{1}{n} \sum_{i=1}^n \Pr(a_i z_i^2 = x).
\]

One can think of the simulation of a sample \( x \sim F \) as a two-stage process. First, we pick an \( i \in [n] \) uniformly at random (this gives a corresponding \( a_i \)), and then we pick \( z \sim \mathcal{N}(0,1) \) independently. The product \( a_i z^2 \) then has distribution given by \( F \).
Suppose we pick \( m = 10n \log(n) \) samples as follows: first we pick \( m \) times independently, uniformly at random from \([n]\) (with replacement) to obtain the set \( Y = \{y_1, \ldots, y_m\} \); then we pick \( m \) independent standard Gaussian random variables \( \{z_1, \ldots, z_m\} \), and finally compute component-wise products \( \{y_1z_1^2, \ldots, y_mz_m^2\} \). Let \( Y_1, \ldots, Y_n \) be a partition of \( Y \) according to which \( a_i \) is assigned to each \( y_j \), i.e., \( Y_i \) is the set of \( y_j \)’s for which \( a_i \) was chosen.

The following bounds follow from standard Chernoff-Hoeffding bounds. For i.i.d. Bernoulli \( \{0,1\} \) random variables with bias \( p \):

\[
\Pr \left( \frac{1}{m} \sum_{i=1}^{m} X_i \geq (1 + \delta)pm \right) \leq \exp \left( -\frac{\delta^2 pm}{3} \right) \tag{2}
\]

Claim 4.2.

\[ \Pr (\exists i : |Y_i| = 0) \leq \frac{1}{n^9} \quad \text{and} \quad \Pr (\exists i : |Y_i| > 40 \ln n) \leq \frac{1}{n^2}. \]

We now assume the above two events do not occur which happens with probability at least \( 2/n^2 \). Next, we draw a subsample of size \( n \) from the set \( Y = \{y_1z_1^2, \ldots, y_mz_m^2\} \) to form the set \( S \). To do so, we simply pick a single representative uniformly at random from each \( Y_i \). From the claim above, we know that each bucket has at least one element, and at most \( 40 \log(n) \) elements in it.

The set \( W \) is the set of values \( y_i z_i^2 \) associated with the \( n \) representatives we picked uniformly at random. A simple observation is that \( W \) is distributed exactly as the \( \{p_1(z_1), \ldots, p_n(z_n)\} \) in the statement of this theorem. In fact, each \( a_i \) shows up exactly once in \( W \), and is multiplied by \( z^2 \) for \( z \sim N(0,1) \), and all these random variables are independent.

Next, we condition on the event that \( \arg \max_i y_i z_i^2 \) and \( \arg \min_i y_i z_i^2 \) are picked in \( W \). This occurs with probability at least \( 1/1600 \log(n)^2 \) since no bucket is of size greater than \( 40 \log(n) \) by Claim 4.2. With this assumption, it is clear that \( \maxgap(W) \geq \maxgap(y_1, \ldots, y_m) \). Thus, it suffices for us to analyse \( \maxgap(y_1z_1^2, \ldots, y_mz_m^2) \). Since the latter random variable is independent of which \( y_j \) are picked for \( W \), we have a reduction from our original random variable model \( \{a_1 z_1^2, \ldots, a_n z_m^2\} \) to (slightly more) samples from a mixture model \( F \).

To lower bound the maximum gap, observe that the density \( F(x) \) is continuous, has its maximum at \( x = 0 \) and monotonically decays to 0 as \( x \to \infty \), since this is true for each of the component distributions \( f_i \). We will now find thresholds \( t_0, t_1 \) such that \( t_1 - t_0 \) is large, and there is good probability that no element of \( W \) takes its value in the interval \( [t_0, t_1] \) and at least one element of \( W \) takes its value to the right of \( t_1 \). We will use the following standard Gaussian tail bound.

Fact 4.3. For \( z \) drawn from \( N(0, 1) \), and \( t \in \mathbb{R} \),

\[
\left(1 - \frac{1}{x^3}\right) e^{-x^2/2} \leq \sqrt{2\pi} \Pr(z \geq x) \leq \frac{1}{x} e^{-x^2/2}.
\]

For some \( t_0 \) and \( t_1 \), we must have that:

\[
\Pr(x \geq t_0) = \sqrt{\frac{2}{\pi}} \cdot \frac{1}{n} \quad \text{and} \quad \Pr(x \geq t_1) = \sqrt{\frac{2}{\pi}} \cdot \frac{1}{2 \sqrt{2n \log(2n)}}
\]
Let us expand out the probabilities explicitly:

\[
\Pr(x \geq t_0) = \frac{1}{n} \sum_{i=1}^{n} \Pr(a_i z_i^2 \geq t_0) = \frac{2}{n} \sum_{i=1}^{n} \Pr(z_i \geq \sqrt{\frac{t_0}{a_i}}) \leq \frac{2}{n\sqrt{2\pi}} \sum_{i=1}^{n} \sqrt{\frac{a_i}{t_0}} \exp\left(-\frac{t_0}{2a_i}\right)
\]

where we used the upper bound on the Gaussian tail. Combining this with the definition of \(t_0\), this implies that there exists some \(i\) for which,

\[
\sqrt{\frac{a_i}{t_0}} \exp\left(-\frac{t_0}{2a_i}\right) \geq \frac{1}{n}.
\]

It follows that for this \(i\), we must have \(t_0 \geq a_i\) and

\[-\frac{t_0}{2a_i} \geq -\log(n)\) or \(t_0 \leq 2a_i \log(n)\).

Using similar reasoning for \(t_1\), with the lower bound on the Gaussian tail, we have

\[
\Pr(x \geq t_1) \geq \frac{2}{n\sqrt{2\pi}} \sum_{i=1}^{n} \left[\left(\frac{a_i}{t_1}\right)^{1/2} - \left(\frac{a_i}{t_1}\right)^{3/2}\right] \exp\left(-\frac{t_1}{2a_i}\right).
\]

Then there must be one term \(i\) for which,

\[
\left[\left(\frac{a_i}{t_1}\right)^{1/2} - \left(\frac{a_i}{t_1}\right)^{3/2}\right] \exp\left(-\frac{t_1}{2a_i}\right) \leq \frac{1}{2\sqrt{2n} \log(n)}
\]

This implies that \(t_1 \geq 2a_i \log(2n)\). Thus, \(t_1 - t_0 \geq \min_i 2a_i(\log 2n - \log n) = 2\log 2 \min_i a_i\).

We can bound the probability of the intervals \([t_0, t_1]\) and \([t_1, \infty)\) by

\[
\Pr(x \in [t_0, t_1]) \leq \Pr(x \geq t_0) \leq \sqrt{\frac{2}{\pi n}}.
\]

and \(\Pr(x \geq t_1)\) respectively. Then by applying the Chernoff bound \([2]\), we obtain that with probability at least \(1 - 1/n^{5/6}\), there are at most \((20/\sqrt{\pi})\log n\) points in the interval \([t_0, t_1]\) and there are at least \((2/\sqrt{\pi})\log n\) points beyond \(t_1\). Since the interval is of at least \(2\log 2 \min_i a_i\), by averaging, there exists an adjacent pair which are at least

\[
\sqrt{\frac{\pi \log 2}{10 \log n}} \min_i a_i
\]

apart as claimed.

To compute the failure probability, we note that \(1/(1600 \log^2 n)\) is far larger than the \(1/n^c\) terms elsewhere in the calculation, so we can bound the failure probability by \(1/(2000 \log^2 n)\). \qed
4.2 Sample complexity and error analysis

The analysis of the algorithm uses a version of the \( \sin(\theta) \) theorem of Davis and Kahan [4]. Roughly speaking, the largest eigenvalue gap controls the magnitude of the error in each subspace \( V_1 \) and \( V_2 \) in the algorithm, each recursive step subsequently accumulates error accordingly, and we have to solve a nonlinear recurrence to bound the total error. We will use the following theorems in the proof. The first is a form of Wedin’s Theorem from [20].

**Theorem 4.4** ([20]). Let \( A, E \in \mathbb{C}^{m \times n} \) be complex matrices with \( m \geq n \). Let \( A \) have singular value decomposition

\[
A = [U_1 U_2 U_3] \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{pmatrix} [V_1^* V_2^*]
\]

and similarly for \( \tilde{A} = A + E \) (with conformal decomposition using \( \tilde{U}_1, \tilde{\Sigma}_1 \) etc). Suppose there are numbers \( \alpha, \beta > 0 \) such that

1. \( \min \sigma(\tilde{\Sigma}_1) \geq \alpha + \beta \)
2. \( \max \sigma(\Sigma_2) \leq \alpha \).

Then,

\[
\| \sin(\Phi) \|_2, \| \sin(\Theta) \|_2 \leq \frac{\| E \|_2}{\beta}
\]

where \( \Phi \) is the (diagonal) matrix of canonical angles between the ranges of \( U_1 \) and \( \tilde{U}_1 \) and \( \Theta \) denotes the matrix of canonical angles between the ranges of \( U_2 \) and \( \tilde{U}_2 \).

We will also need Taylor’s theorem with remainder.

**Theorem 4.5.** Let \( f : \mathbb{R} \to \mathbb{R} \) be a \( C^n \) continuous function over some interval \( I \). Let \( a, b \in I \), then

\[
f(b) = \sum_{k=1}^{n-1} \frac{f^{(k)}(a)}{k!} (b - a)^k + \frac{f^{(n)}(\xi)}{n!} (b - a)^n,
\]

for some \( \xi \in [a, b] \).

The following simple bounds will be used in estimating the sample complexity.

**Lemma 4.6.** Suppose that the random vector \( x \in \mathbb{R}^n \) is drawn from an isotropic distribution \( F \). Then for \( 1 \leq j \leq n \),

\[
\text{Var}(x_j e^{iu^T x}) \leq 1, \quad \text{Var}(x_j^2 e^{iu^T x}) \leq \mathbb{E} \left( x_j^4 \right) \quad \text{and} \quad \text{Var}(x_i x_j e^{iu^T x}) \leq 1 \text{ for } i \neq j.
\]

The last is Theorem 1.2 from [21].

**Theorem 4.7** ([21]). Consider a random vector \( x \in \mathbb{R}^n \) with covariance \( \Sigma \), such that \( \| x \| \leq \sqrt{m} \) almost surely. Let \( \epsilon \in (0, 1) \) and \( t \geq 1 \), then with probability at least \( 1 - 1/n^{t/2} \), if \( N \geq C(t/\epsilon)^2 \| \Sigma \|^{-1} m \log(n) \), then \( \| \Sigma_N - \Sigma \| \leq \epsilon \| \Sigma \| \).
Proof of Theorem 1.2. First, we prove that when we run the algorithm and compute a set of eigenvalues, that there exists at least one large gap in the set \( \{ \text{Re}(g_j(t_j)) \} \), the diagonal entries in the decomposition of \( \text{Re}(\Sigma_u) \). For this, we recall that \( g_j = \psi_j'' \), and using the Taylor expansion of \( \psi_j \), we write each \( g_j \) as follows.

\[
g_j(t_j) = -\sum_{l=2}^{k} \text{cum}(s_j)(it_j)^{l-2} \frac{(it_j)^{k-1}}{(l-2)!} - g^{(k+1)}(\xi)(it_j)^{k-1} \frac{(k-1)!}{(k-1)!}
\]

where \( \xi \in [0, t_j] \) and \( p_i \) is a polynomial of degree \((k - 2)\). Using \( k = 4 \) and \( j = 1 \),

\[
g_1(t_1) = -1 - \text{cum}_3(s_1)(it_1) - \text{cum}_4(s_1)(it_1)^2 + R_1(t_1)(it_1)^3.
\]

When we take the real part of the matrix in Step 6 of the algorithm, we can discard the pure imaginary term arising from the first cumulant. We must retain the error term as we do not know a priori whether the error derivative term has a complex component or not. Truncating after the second order terms, this gives a family of polynomials

\[
p_j(t_j) = -1 + \text{cum}_{4}(s_j)\frac{t_j^2}{2}.
\]

Since \( \text{cum}_{4}(s_j) \geq \Delta \) and \( t_j \) is drawn from \( N(0, \sigma^2) \), we can now apply Theorem 4.1 that shows that with probability \( 1/2000 \log^2 n \),

\[
\max \text{gap}(p_j(t_j)) \geq \frac{\log 2}{20 \log n} \Delta \sigma^2.
\]

Thus with \( 8000 \log^3 n \) different random vectors \( u \), with probability at least \( 1 - (1/n^2) \) we will see a gap of at least this magnitude.

Next, we bound the remainder. Using Lemmas 4.9 and 10.1 from [12], for \( t_j \in [-1/4, 1/4] \), we have

\[
|R_j(t_j)| \leq \frac{4!2^4 \mathbb{E}(|s_j|^5)}{(3/4)^5}.
\]

So the full remainder term with probability at least \( 1 - (1/n^2) \) is at most

\[
|R_j(t_j)| \frac{3^3}{3!} \leq \frac{4^7}{3^5} \mathbb{E}(|s_j|^5) \frac{|t_j|^3}{3!} \leq \frac{4^7}{3^5} \mathbb{E}(|s_j|^5) \sigma^3 (4 \log n)^{3/2} \leq \frac{\log 2}{40 \log n} \Delta \sigma^2
\]

for

\[
\sigma \leq \frac{\Delta}{1000(\log^{5/2} n)\mathbb{E}(|s_j|^5)}.
\]

Next, partition the eigenvectors according to which side of the maximum gap they fall on, let \( V \) and \( V^\perp \) denote these sets respectively. We bound the error using Theorem 4.4 which bounds the canonical angles in terms of the gap. Suppose that in each iteration, we take enough samples so that the empirical version of \( D^2 \log(\phi)_u \) is within \( \epsilon' \) of the true one. Then applying the theorem yields that the for the subspaces spanned by \( V \) and \( W = V^\perp \), that there exists a partition of the
columns of $A$ (which we may take, without loss of generality, to be ordered appropriately) such that:

$$\|\sin(\theta(V, \{A_1, \ldots, A_k\})\| \leq \frac{\epsilon'}{\Delta \sigma^2 \log 2} = \frac{20\epsilon' \log n}{\Delta \sigma^2 \log 2}.$$ 

Now consider the call of Recursive FPCA on the subspace $V$ of dimension $k$. In this subspace, we can write the Hessian matrix as:

$$D^2 \log(\phi) = (V^T[A_1, \ldots, A_k])\text{diag}(\lambda_1, \ldots, \lambda_k) (V^T[A_1, \ldots, A_k])^T$$

Note that by definition, we have that $\sin(\theta) = V^T[A_{k+1}, \ldots, A_n]$, thus the second term is upper bounded by $(20\epsilon' \log n / \log(2) \Delta)^2$; we must also add the sampling error from the second iteration (say another $\epsilon'$). Suppose that we write the recurrence for the overall error $E_k$ at a recursive call at depth $k$, then:

$$E_k = \epsilon' + \left(\frac{E_{k-1} \log(n)}{c \Delta \sigma^2}\right)^2.$$ 

For small $\epsilon'$ (to be determined presently), we can simply use the following recurrence to bound the total error by $2\epsilon'$. The claim follows by induction on $i$.

**Claim 4.8.** Fix $a, b > 0$ where $4a/b^2 \leq 1$, and define the recurrence $y_{i+1} = a + (y_i/b)^2$ and $y_0 = 0$, then $y_i \leq 2a$ for all $i$.

**Proof.** We proceed via induction. Clearly this is true for $i = 0$. Now suppose that it is true for $i \leq k$. Then

$$y_{i+1} = a + (y_i/b)^2 \leq a + 4a^2/b^2 \leq 2a.$$ 

In the terminal nodes of the recurrence, this gets blown up to $40 \log(n)\epsilon'/\Delta \sigma^2 \log 2$; in this iteration the output error will give the overlap between the output vectors and those of $A$. Thus, with $M_5 = \max_i \mathbb{E}\left(|s_i|^5\right)$, setting

$$\epsilon' = \frac{\epsilon \Delta \sigma^2 \log 2}{40 \log n} = \frac{\epsilon \Delta^3 \log 2}{40(10 \log n)^6 M_5^5}$$

suffices to give total error at most $\epsilon$.

For the sample complexity, we have to take enough samples so that for $8000 \log^3 n$ different instantiations of the Fourier derivative matrix, the spectral norm error is within $\epsilon'$ with high probability. For this, we apply Thm. 4.7, after first separating the real and imaginary parts of the second derivative matrix, and then splitting these further into positive and negative parts. It suffices to estimate three matrix-valued random variables $\mathbb{E}(xx^T \exp(iu^T x))$, $\mathbb{E}(x \exp(iu^T x))$ and $\mathbb{E}(\exp(iu^T x))$. The latter two are easy to estimate using $O(n)$ samples by applying Lemma 4.6. Thus, it suffices for us to show that we can estimate the second order term $\mathbb{E}(xx^T \exp(iu^T x))$ using
only a nearly linear number of samples. This will follow from the assumption that $\|s\| \leq K \sqrt{n}$ via Vershynin’s theorem. We rewrite this term into four easily estimable parts:

$$
\mathbb{E} (xx^T \exp(\imath u^T x)) = \mathbb{E} (xx^T \cos(u^T x)) + \imath \mathbb{E} (xx^T \sin(u^T x))
$$

$$
= \mathbb{E} \left( xx^T 1_{\cos(u^T x) \geq 0} \cos(u^T x) \right) + \mathbb{E} \left( xx^T 1_{\cos(u^T x) < 0} \cos(u^T x) \right)
+ \imath \mathbb{E} \left( xx^T 1_{\sin(u^T x) \geq 0} \sin(u^T x) \right) + \imath \mathbb{E} \left( xx^T 1_{\sin(u^T x) < 0} \sin(u^T x) \right)
$$

Let us estimate these four quantities using independent samples. If each one is within $\epsilon/4$, then we can estimate $\mathbb{E} (xx^T \exp(\imath u^T x))$ to within $\epsilon$ in the spectral norm. Consider, for example, the first term $xx^T 1_{\cos(u^T x) \geq 0} \cos(u^T x)$, then we can define the random vector $y = x 1_{\cos(u^T x) \geq 0} \sqrt{\cos(u^T x)}$. Then it is clear that:

$$
yy^T = xx^T 1_{\cos(u^T x) \geq 0} \cos(u^T x)
$$

In particular, observe that $0 \leq \mathbb{E} ((u^T y)^2) \leq \mathbb{E} ((u^T x)^2) \leq 1$ for all unit vectors $u$. Thus, we must have that the eigenvalues of $\mathbb{E} (yy^T)$ are all bounded by 1. Note also, that $\|y\| \leq \sqrt{m}$ if this is in fact the case for $x$ as well. Now, we apply Theorem 1.2 from [21] to $y$: by hypothesis, we can take $m = K^2 n$ and $t = 2$. Next, we shall use $N$ samples for the entire algorithm (without resampling), and simply apply the union bound against a failure probability of $1/n^2$, thereby giving us a high probability statement. To achieve error $\epsilon$, the sample complexity is

$$
O \left( n \cdot \frac{K^2 M_5^2 \log^{13} n}{\Delta^6 \epsilon^2} \right).
$$

\[ \Box \]

5 Conclusion

Our work was motivated by experiments on Fourier PCA and tensor-based methods, which appear to need a rather large number of samples even for modest values of the dimension $n$. The recursive variant presented here scales smoothly with the dimension, and is available as MATLAB code [22].

While for our algorithm here, it suffices to consider quadratic polynomials, analyzing the gaps of a family of polynomials over Gaussians is an interesting problem on its own. One surprise here is that even for degree 3, the polynomial $p(x) = x(x-a)(x+a)$ where $a = \sqrt{2 \log(n)}$ evaluated at $n$ random points from $N(0,1)$ has maximum gap only $O(1/n^{0.6})$. However, it is unclear whether this polynomial corresponds to a legal characteristic polynomial. Thus, an important open question is whether the maximum spacing ideas can be extended beyond degree 2.

References

[1] Anandkumar, A., Foster, D., Hsu, D., Kakade, S., and Liu, Y.-K. A spectral algorithm for latent dirichlet allocation. In Advances in Neural Information Processing Systems 25 (2012), pp. 926–934.
[2] Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. CoRR abs/1210.7559 (2012).

[3] Arora, S., Ge, R., Moitra, A., and Sachdeva, S. Provable ICA with unknown gaussian noise, with implications for gaussian mixtures and autoencoders. In NIPS (2012), pp. 2384–2392.

[4] Belkin, M., Rademacher, L., and Voss, J. Blind signal separation in the presence of Gaussian noise. In Proc. of COLT (2013).

[5] Cardoso, J. Source separation using higher order moments. In International Conference on Acoustics, Speech, and Signal Processing (1989).

[6] Comon, P., and Jutten, C., Eds. Handbook of Blind Source Separation. Academic Press, 2010.

[7] Davis, C., and Kahan, W. M. The rotation of eigenvectors by a perturbation III. SIAM Journal on Numerical Analysis 7, 1 (1970), 1–46.

[8] Deheuvels, P. Strong limit theorems for maximal spacings from a general univariate distribution. The Annals of Probability (1984), 1181–1193.

[9] Deheuvels, P. The limiting behaviour of the maximal spacing generated by an iid sequence of gaussian random variables. Journal of applied probability (1985), 816–827.

[10] Deheuvels, P. On the influence of the extremes of an iid sequence on the maximal spacings. The Annals of Probability (1986), 194–208.

[11] Frieze, A. M., Jerrum, M., and Kannan, R. Learning linear transformations. In FOCS (1996), pp. 359–368.

[12] Goyal, N., Vempala, S., and Xiao, Y. Fourier PCA and robust tensor decomposition. In Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014 (2014), pp. 584–593.

[13] Guedon, O., and Rudelson, M. $L_p$ moments of random vectors via majorizing measures. Advances in Mathematics 208 (2007), 798–823.

[14] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. The elements of statistical learning, vol. 2. Springer, 2009.

[15] Hillar, C., and Lim, L.-H. Most tensor problems are NP-hard. Journal of the ACM 60 (2013).

[16] Hsu, D., and Kakade, S. M. Learning mixtures of spherical Gaussians: moment methods and spectral decompositions. In ITCS (2013), pp. 11–20.

[17] Hyvärinen, A., Karhunen, J., and Oja, E. Independent Component Analysis. Wiley, 2001.

[18] Ngiam, J., Chen, Z., Chia, D., Koh, P. W., Le, Q. V., and Ng, A. Y. Tiled convolutional neural networks. In Advances in Neural Information Processing Systems (2010), pp. 1279–1287.
[19] Nguyen, P. Q., and Regev, O. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology 22, 2 (2009), 139–160.

[20] Stewart, G. W., and Sun, J.-G. Matrix perturbation theory. Academic press, 1990.

[21] Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. In Compressed Sensing, Theory and Applications, Y. Eldar and G. Kutyniok, Eds. Cambridge University Press, Oxford, 2010, pp. 210–268.

[22] Xiao, Y. Fourier pca package. GitHub (2014). http://github.com/yingusxiaous/libFPCA.

[23] Yeredor, A. Blind source separation via the second characteristic function. Signal Processing 80, 5 (2000), 897–902.