JMLR: Workshop and Conference Proceedings vol 40:1–14, 2015

Max vs Min: Tensor Decomposition and ICA with nearly Linear Sample Complexity

Santosh S. Vempala VEMPALA@GATECH.EDU
School of Computer Science, Georgia Tech

Ying Xiao YING.XIAO@GATECH.EDU
School of Computer Science, Georgia Tech

Abstract

We present a simple, general technique for reducing the sample complexity of matrix and tensor decomposition algorithms applied to distributions. We use the technique to give a polynomial-time algorithm for standard ICA with sample complexity nearly *linear* in the dimension, thereby improving substantially on previous bounds. The analysis is based on properties of random polynomials, namely the spacings of an ensemble of polynomials. Our technique also applies to other applications of tensor decompositions, including spherical Gaussian mixture models.

Keywords: Independent Component Analysis, Tensor decomposition, Fourier PCA, Sample Complexity, Eigenvalue spacing

1. Introduction

Matrix and tensor decompositions have proven to be a powerful theoretical tool in a number of models of unsupervised learning, e.g., Gaussian mixture models (Vempala and Wang, 2004; Anderson et al., 2013; Hsu and Kakade, 2013; Goyal et al., 2014), Independent Component Analysis (Frieze et al., 1996; Nguyen and Regev, 2009; Belkin et al., 2013; Arora et al., 2012; Goyal et al., 2014), and topic models (Anandkumar et al., 2012b) to name a few recent results. A common obstacle for these approaches is that while the tensor-based algorithms are polynomial in the dimension, they tend to have rather high sample complexity. As data abounds, and labels are expensive, efficient methods for unsupervised learning — recovering latent structure from unlabeled data — are becoming more important.

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 (Hastie et al., 2009). More recently, it has been used as a tool for sparsifying layers in deep neural nets (Ngiam et al., 2010). 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 $s_1, \dots, s_n \in \mathbb{R}$ 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 ϵ. 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 substantially on previous polynomial-time algorithms (Frieze et al., 1996; Nguyen and Regev, 2009; Arora et al., 2012; Belkin et al., 2013; Anandkumar et al., 2012b; Goyal et al., 2014), which all require a higher polynomial number of samples ($\Omega(n^5)$ or higher). Our technique also applies more broadly to tensor decomposition problems where the tensor is estimated from samples. In particular, it applies to tensor decomposition methods used in analyzing topic models and learning mixtures of spherical Gaussians with linearly independent means; it extends to the setting where the model is corrupted with Gaussian noise. Before stating our results precisely, we place it in the context of related work.

1.1. Related work

Tensor decomposition is a major technique in the literature on ICA. The latter is vast, with many proposed algorithms (see Comon and Jutten (2010); Hyvärinen et al. (2001)). Frieze et al. (1996) 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 (Nguyen and Regev, 2009; Arora et al., 2012; Belkin et al., 2013; Anandkumar et al., 2012b). 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. (2014) recently gave an algorithm that can deal with differences from being Gaussian at any moment. 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 all these papers can be viewed as 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 straightforward, and many versions of this decomposition lead directly to NP-hard problems (Hillar and Lim, 2013; Brubaker, 2009). The application of tensor decomposition to ICA was proposed by Cardoso (1989). Such decompositions were used by Anandkumar et al. (2012a) and Hsu and Kakade (2013) to give provable algorithms for various latent variable models. Goyal et al. (2014) 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.

1.2. Our technique and results

Suppose we are interested in finding the eigenvectors of a matrix estimated from samples, e.g., a covariance matrix of a distribution in $\mathbb{R}^n$. Then, for the decomposition to be unique, we need the eigenvalues to be distinct, and the sample complexity grows with the inverse square of the *minimum* eigenvalue gap. In many situations however, for such matrices, the *maximum* eigenvalue gap is much larger. The core idea of this paper is to use as many samples as needed to estimate the largest gap in the eigenvalues accurately, find the corresponding subspaces (which will be stable due the large gap), then recurse in the two subspaces by projecting samples to them. Our target application will be to ICA, but the technique can be used for tensor decomposition in other settings.

Our main result is a polynomial-time algorithm for ICA using only a nearly linear number of samples. Since each column of A can only be recovered up to a scaling of the column, we can assume w.l.o.g. that s is isotropic.

Theorem 1 *Let $x \in \mathbb{R}^n$ be given by an ICA model $x = As$, where $A \in \mathbb{R}^{n\times n}$, the components of s are independent, $\|s\| \le K\sqrt{n}$ almost surely, and for each i, $\mathbb{E}\left(s_i\right) = 0, \mathbb{E}\left(s_i^2\right) = 1$, and $|\mathrm{cum}_4(s_i)| = \left|\mathbb{E}\left(|s_i|^4\right) - 3\right| \ge \Delta$. Let $\mathbf{M} \ge \max_i \mathbb{E}\left(|s_i|^5\right)$. Then for any $\epsilon < \Delta^3/(10^8\mathbf{M}^2\log^3 n)$, with $\sigma = \Delta/(1000\mathbf{M}^2\log^{3/2} n)$, with high probability,* ***Recursive FPCA*** *finds vectors $\{b_1, \dots, b_n\}$ such that there exist signs $\xi_i = \pm 1$ satisfying $\left\|A^{(i)} - \xi_i b_i\right\| \le \epsilon\|A\|_2$ for each column $A^{(i)}$ of A, using*

$$O\left(n \cdot \frac{K^2\mathbf{M}^4\log^7 n}{\Delta^6\epsilon^2}\right) = O^*(n)$$

samples. The running time is bounded by the time to compute $\tilde{O}(n)$ Singular Value Decompositions on real symmetric matrices of size $n \times n$.

This improves substantially on the previous best sample complexity. The algorithm is a *recursive* variant of standard tensor decomposition. It proceeds by first puting the distribution in near-isotropic position, then computing the eigenvectors of a reweighted covariance matrix; these eigenvectors are essentially the columns $A^{(i)}$. To do so accurately with few samples requires that the eigenvalues of the random matrix are well-spaced. To estimate all the eigenvectors, we need large spacings between all $n-1$ adjacent eigenvalue pairs, i.e., 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.

The idea behind our new algorithm is very simple: instead of estimating all all the gaps (and eigenvectors) accurately in one shot, we group the eigenvectors according to which side of the *largest* gap $\max_i \lambda_{i+1} - \lambda_i$ they fall. The vectors *inside* either of these subspaces are not necessarily close to the desired $A^{(i)}$, but we can proceed recursively in each subspace, *re-using 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 needed 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. For the tensor approach to ICA, the complexity goes down to $\tilde{O}(n^2)$. To go all the way to linear, we will apply the maxgap idea and recursion to the Fourier PCA algorithm of Goyal et al. (2014).

This paper has three components of possibly general interest: 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 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 η is from an unknown (not necessarily spherical) Gaussian distribution. We do not treat these extensions in detail here as they are similar to Goyal et al. (2014), but with the improved sample complexity of the core algorithm. Our approach can also be used to improve the sample complexity of other applications of tensor methods (Anandkumar et al., 2012b), including learning hidden Markov models and latent topic models.

2. Outline of approach

Given a tensor $T = \sum_{i=1}^{n} \alpha_i v_i \otimes v_i \otimes v_i \otimes v_i$, we consider the matrix

$$T(u,u) = \sum_{i=1}^{n} \alpha_i (v_i \cdot u)^2 v_i \otimes v_i = V \text{diag}\left(\alpha_i (v_i \cdot u)^2\right) V^T.$$

If u is chosen randomly, then with high probability, the diagonal entries will be distinct. For a Gaussain u, the diagonal entries will themselves be independent Gaussians, and the minimum gap will be $O(1/n)$. However, the *maximum* gap will be much larger (inverse logarithmic), and we exploit this in the following algorithm. The same approach applies to tensors of order k, where we have $T(u, u, \ldots, u)$ with $k-2$ arguments and the RHS coefficients are $\alpha_i (v_i \cdot u)^{k-2}$. We describe the algorithm below for fourth order tensors ($k = 4$), but an almost identical algorithm works for any $k \geq 3$, with $T(u,u)$ replaced by $T(u, \ldots, u)$ with $k-2$ copies of u as arguments. To apply the algorithm for a specific unsupervised learning model, we just have to define the tensor T appropriately (Anandkumar et al., 2012b). For ICA, this is

```
Recursive-Decompose($T$, Projection matrix $P \in \mathbb{R}^{n \times \ell}$)
    1. (Gaussian weighting) Pick a random vector $u$ from $N(0,1)^n$.
    2. (SVD) Compute the spectral decomposition of $P^T T(u,u) P$ to obtain $\{\lambda_i\}$, $\{v_i\}$.
    3. (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 $V_2 = \{v_{i+1}, \ldots, v_\ell\}$.
    4. (Recurse) For $j = 1, 2$: if $|V_j| = 1$ set $W_j = V_j$, else $W_j =$ Recursive-Decompose$(T, PV_1)$.
    5. Return $[W_1 \quad W_2]$.
```

$$T = \mathbb{E}\left(x \otimes x \otimes x \otimes x\right) - M$$

where $M_{ijkl} = \mathbb{E}\left(x_i x_j\right) \mathbb{E}\left(x_k x_l\right) + \mathbb{E}\left(x_i x_k\right) \mathbb{E}\left(x_j x_l\right) + \mathbb{E}\left(x_i x_l\right) \mathbb{E}\left(x_j x_k\right)$. As we show in Theorem 3, the maximum gap grows as $\Omega(1)$, while the minimum gap is $O(1/n^2)$. While this already improves the known sample complexity bounds to quadratic in the dimension, it is still $\Omega(n^2)$.

2.1. Fourier PCA

To achieve near-linear sample complexity for ICA, we will apply the recursive decomposition idea to the Fourier PCA approach of Goyal et al. (2014). 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

$$\phi_j(u_j) = \mathbb{E}\left(e^{iu_j s_j}\right) \quad \text{and} \quad \psi_i(u_j) = \log(\phi_j(u_j)) = \sum_{k=1}^{\infty} \mathrm{cum}_k(s_j) \frac{(iu_j)^k}{k!}. \tag{1}$$

Here $\mathrm{cum}_k(y)$ 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 $D^2\psi$:

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

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

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

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

$$D^2\psi = \mathrm{diag}\left(\psi_j''(u_j)\right) = \mathrm{diag}\left(g_j(u_j)\right).$$

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 (2000). 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 *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*

$$D^2\psi = A\mathrm{diag}\left(g_i((A^T u))_i\right) A^T.$$

To obtain a robust algorithm, we need the eigenvalues of $D^2\psi$ being adequately spaced (so that the error arising from sampling does not mix the columns of A). For this, Goyal et al. (2014) pick a random vector $u \sim N(0, \sigma^2 I_n)$, so that the $g_i(u_i)$ are sufficiently anti-concentrated with σ small enough and the number of samples large enough so that with high probability for all pairs $i \neq j$, they could guarantee $\left|g_i((A^T u)_i) - g_j((A^T u)_j)\right| \geq \delta$ for a suitable δ, leading to a (large) polynomial complexity.

3. Recursive Fourier PCA

We partition the eigenvectors into two sets according to where their eigenvalues fall relative to the maximum gap, project to the two subspaces spanned by these sets and recurse, *re-using the initial sample*. The parameter σ below can be safely set according to Theorem 1 but in practice we suggest starting with a constant σ and halving it till the output of the algorithm has low error (which can be checked against a new sample). The recursive decomposition step can be carried out simultaneously for all the decomposed blocks. In other words, viewing the decomposition as a tree, the next step can be carried for all nodes at the same level, with a single SVD, and a single vector $u \sim N(0,1)^n$. This is the same as doing each block separately with the vector u projected to the span of the block.

```
Recursive FPCA($\sigma$, Projection matrix $P \in \mathbb{R}^{n \times k}$)
  1. (Isotropy) Find an isotropic transformation $B^{-1}$ with
       $B^2 = \frac{1}{|S|} \sum_{x \in S} P^T (x - \bar{x})(x - \bar{x})^T P.$
  2. (Fourier weights) Pick a random vector $u$ from $N(0, \sigma^2 I_n)$. For every $x$ in the sample $S$,
     compute $y = B^{-1} P^T x$, and its Fourier weight
       $w(y) = \frac{e^{iu^T x}}{\sum_{x \in S} e^{iu^T x}}.$
  3. (Reweighted Covariance) Compute the covariance matrix of the points $y$ reweighted by $w(y)$
       $\mu_u = \frac{1}{|S|} \sum_{y \in S} w(y) y$   and   $\Sigma_u = -\frac{1}{|S|} \sum_{y \in S} w(y)(y - \mu_u)(y - \mu_u)^T.$
  4. (SVD) Compute the spectral decomposition $\{\lambda_i\}$, $\{v_i\}$ of $\mathrm{Re}(\Sigma_u)$.
  5. (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 $V_2 = \{v_{i+1}, \ldots, v_k\}$.
  6. (Recursion) For $j = 1, 2$: if $|V_j| = 1$ set $W_j = V_j$, else $W_j =$ Recursive FPCA($\sigma, PV_j$).
  7. Return $[W_1 \quad W_2]$.
```

4. 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 ϵ 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:

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

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

Theorem 3 *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 $1/(2000 \log^2 n)$,*

$$\operatorname{maxgap}(p_1(z_1), \ldots, p_n(z_n)) \geq \frac{1}{50} \min_i a_i.$$

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 Deheuvels (1985, 1984, 1986). 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 $z \sim N(0, 1)$ 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 $Y = \{y_1, \dots, y_m\}$; then we pick m independent standard Gaussian random variables $\{z_1, \dots, z_m\}$, and finally compute component-wise products $\{y_1 z_1^2, \dots, y_m z_m^2\}$. Let $Y_1, \dots, Y_n$ 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:

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

Claim 4 $\mathsf{Pr}\left(\exists i : |Y_i| = 0\right) \leq \frac{1}{n^9}$ *and* $\mathsf{Pr}\left(\exists i : |Y_i| > 40\log n\right) \leq \frac{1}{n^2}$.

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 $Y = \{y_1 z_1^2, \dots, y_m z_m^2\}$ 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 $\{p_1(z_1), \dots, p_n(z_n)\}$ 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. With this assumption, it is clear that $\text{maxgap}\left(W\right) \geq \text{maxgap}\left(y_1, \dots, y_m\right)$. Thus, it suffices for us to analyse $\text{maxgap}\left(y_1 z_1^2, \dots, y_m z_m^2\right)$. 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 $\{a_1 z_1^2, \dots, a_m z_m^2\}$ 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 pick 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 pick t_0 s.t.

$$\mathsf{Pr}\left(x \geq t_0\right) = \frac{1}{n\log(n)}$$

and $t_1 = t_0 + 2\min_i a_i$. Let $a_1 = \min_i a_i$. With these settings, we have,

$$\frac{\mathsf{Pr}\left(x \geq t_1\right)}{\mathsf{Pr}\left(x \geq t_0\right)} = \frac{\sum_{i=1}^{n}\mathsf{Pr}\left(a_i z_i^2 \geq t_1\right)}{\sum_{i=1}^{n}\mathsf{Pr}\left(a_i z_i^2 \geq t_0\right)} \geq \min_i \frac{\mathsf{Pr}\left(a_i z_i^2 \geq t_1\right)}{\mathsf{Pr}\left(a_i z_i^2 \geq t_0\right)} = \frac{\mathsf{Pr}\left(a_1 z_1^2 \geq t_1\right)}{\mathsf{Pr}\left(a_1 z_1^2 \geq t_0\right)} = \frac{\mathsf{Pr}\left(z_1 \geq \sqrt{t_1/a_1}\right)}{\mathsf{Pr}\left(z_1 \geq \sqrt{t_0/a_1}\right)}$$

where the second inequality follows by simply expanding the densities explicitly.

Next, we will use the following standard Gaussian tail bound Feller (1968).

Fact 5 *For z drawn from $N(0, 1)$, and $t \in \mathbb{R}$,*

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

From this, we have that $t_0 \geq 2a_1 \ln n$ and

$$\frac{\mathsf{Pr}\left(x \geq t_1\right)}{\mathsf{Pr}\left(x \geq t_0\right)} \geq \frac{\sqrt{\frac{a_1}{t_1}}\left(1 - \frac{a_1}{t_1}\right)}{\sqrt{\frac{a_1}{t_0}}} \cdot \frac{\exp(-t_1/2a_1)}{\exp(-t_0/2a_1)} \geq \frac{1}{2e}.$$

Thus, $\mathsf{Pr}\left(x \geq t_1\right) \geq 1/(2en\log(n))$.

To conclude, observe that with small constant probability, there are at most 100 points $a_i z_i^2$ inside the interval $[t_0, t_1]$, and there exists at least one point to the right of the interval. Thus, there must exist one spacing which is at least $2a_1/100$. The failure probability is dominated by $1-1/(1600\log^2 n)$ as the other terms are of lower order, hence we can bound the failure probability by $1 - 1/(2000\log^2 n)$. ■

For higher order monomials, a slight modification to this argument yields:

Theorem 6 *Let $\{p_1(x), \ldots, p_n(x)\}$ be a set of n degree d polynomials of the form $p_i(x) = a_i x^d$ where $a_i > 0$ for all i and $\{z_1, \ldots, z_n\}$ be iid standard Gaussians. Then, with probability at least $1/(2000\log^2 n)$,*

$$\operatorname{maxgap}\left(p_1(z_1), \ldots, p_n(z_n)\right) \geq \frac{d}{50} \min_i a_i^{\frac{2}{d}} \log(n)^{\frac{1}{2}-\frac{1}{d}}.$$

4.2. Sample complexity and error analysis

The analysis of the algorithm uses a version of the $\sin\theta$ theorem of Davis and Kahan (1970). 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 Stewart and Sun (1990).

Theorem 7 (Stewart and Sun (1990)) *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

$$\min \sigma(\tilde{\Sigma}_1) \geq \alpha + \beta \quad \text{and} \quad \max \sigma(\Sigma_2) \leq \alpha.$$

Then,

$$\left\|\sin(\Phi)\right\|_2, \left\|\sin(\Theta)\right\|_2 \leq \frac{\|E\|_2}{\beta}$$

where Φ is the(diagonal) matrix of canonical angles between the ranges of U_1 and $\tilde{U}_1$ and Θ 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 8 *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 9 *Suppose that the random vector $x \in \mathbb{R}^n$ is drawn from an isotropic distribution F. Then for $1 \le j \le n$,*

$$\mathrm{Var}(x_j e^{iu^T x}) \le 1, \quad \mathrm{Var}(x_j^2 e^{iu^T x}) \le \mathbb{E}\left(x_j^4\right) \textit{ and } \mathrm{Var}(x_i x_j e^{iu^T x}) \le 1 \textit{ for } i \ne j.$$

The next is Theorem 1.2 from Vershynin (2010).

Theorem 10 (Vershynin (2010)) *Consider a random vector $x \in \mathbb{R}^n$ with covariance Σ, such that $\|x\| \le \sqrt{m}$ almost surely. Let $\epsilon \in (0,1)$ and $t \ge 1$, then with probability at least $1 - 1/n^{t^2}$, if $N \ge C(t/\epsilon)^2 \|\Sigma\|^{-1} m \log(n)$, then $\|\Sigma_N - \Sigma\| \le \epsilon \|\Sigma\|$.*

Proof [of Theorem 1] If A is not unitary to begin with, by Theorem 10 a sample of size $O(n \log n/\epsilon_1^2)$ can be used to put x in near-istropic position. After this transformation the matrix $\tilde{A}$ obtained is nearly unitary, i.e., $\|\tilde{A} - \bar{A}\|_2 \le \epsilon'$ where $\bar{A}$ is unitary. Henceforth, we assume that $\bar{A} = A$. 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 $\{\mathrm{Re}(g_j(t_j))\}$, the diagonal entries in the decomposition of $\mathrm{Re}(\Sigma_u)$. We will do this for the unitary matrix A, and it will follow for the estimated matrix $\tilde{A}$, since their eigenvalues are within ϵ'.

We recall that $g_j = \psi_j''$, and using the Taylor expansion of ψ_j (1), we write each g_j as follows.

$$g_j(t_j) = -\sum_{l=2}^{k} \mathrm{cum}_l(s_j)\frac{(it_j)^{l-2}}{(l-2)!} - g^{(k+1)}(\xi)\frac{(it_j)^{k-1}}{(k-1)!} \tag{3}$$

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

$$g_1(t_1) = -1 - \mathrm{cum}_3(s_1)(it_1) - \mathrm{cum}_4(s_1)\frac{(it_1)^2}{2} + R_1(t_1)\frac{(it_1)^3}{3!}.$$

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

$$p_j(t_j) = -1 + \mathrm{cum}_4(s_j)\frac{t_j^2}{2}.$$

Since $\mathrm{cum}_4(s_j) \ge \Delta$ and t_j is drawn from $N(0, \sigma^2)$, we can now apply Theorem 3 that shows that with probability $1/2000 \log^2 n$,

$$\mathrm{maxgap}\left(p_j(t_j)\right) \ge \frac{\Delta\sigma^2}{50}.$$

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 Goyal et al. (2014), for $t_j \in [-1/4, 1/4]$, we have

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

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

$$|R_j(t_j)\frac{(t_j)^3}{3!}| \leq \frac{4^7}{3^5}\mathbb{E}\left(|s_j|^5\right)|t_j|^3 \leq \frac{4^7}{3^5}\mathbb{E}\left(|s_j|^5\right)\sigma^3(4\log n)^{3/2} \leq \frac{1}{100}\Delta\sigma^2$$

for

$$\sigma \leq \frac{\Delta}{1000(\log^{3/2} n)\mathbb{E}\left(|s_j|^5\right)}.$$

Let V and $V^\perp$ denote the sets of eigenvectors above and below the maximum gap respectively. We bound the error using Theorem 7, 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 $D^2\psi(u)$ is within ϵ' of the true one. Then applying the theorem yields that 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:

$$\left\|\sin(\Theta(V, \{A^1, \ldots, A^k\}))\right\| \leq \frac{50\epsilon'}{\Delta\sigma^2}.$$

For Recursive FPCA in the subspace V of dimension k, we can write the matrix as:

$$\begin{aligned} D^2\psi(u) = {} & (V^T[A^1, \ldots, A^k])\operatorname{diag}(\lambda_1, \ldots, \lambda_k)(V^T[A^1, \ldots, A^k])^T \\ & + (V^T[A^{k+1}, \ldots, A^n])\operatorname{diag}(\lambda_{k+1}, \ldots, \lambda_n)(V^T[A^{k+1}, \ldots, A^n])^T \end{aligned}$$

The additional sampling error in this iteration (for the new u) is bounded by ϵ'. By definition, we have that $\sin(\Theta) = V^T[A^{k+1}, \ldots, A^n]$, thus the second term is upper bounded by $(50\epsilon'/\Delta\sigma^2)^2$. For the first term, we can imagine making $V^T T(u,u) V$ isotropic, in which case we must account for the slight nonorthogonality of the columns of V and $A^1, \ldots, A^k$. For this we have to multiply the error by

$$\|(V^T[A^1, \ldots, A^k])^{-1}\|_2 \leq \|\cos(\Theta)^{-1}\| \leq \|I + \sin(\Theta)^2\|$$

where $\sin(\Theta)$ is the error accumulated so far, and we assume that $\|\sin(\Theta)\| \leq 1/2$.

We can write the recurrence for the overall error E_i at a recursive call at depth i, then:

$$E_i \leq (1 + E_{i-1}^2)\left(\epsilon' + \left(\frac{E_{i-1}}{c\Delta\sigma^2}\right)^2\right).$$

We apply Claim 11 (in the appendix) with $a = \epsilon'$ and $b = \Delta\sigma^2/50$. To satisfy the condition it suffices to have $\epsilon' \leq (\Delta\sigma^2/50)^2/8$, which we will achieve by setting $\epsilon' = \epsilon\Delta\sigma^2/100$ since $\epsilon < \Delta^3/(10^8 \mathbf{M}_5^2 \log^3 n)$. In the terminal nodes of the recurrence, the error gets blown up to at most $2\epsilon'$ and this implies a final error between the output vectors and the columns of A of at most $2\epsilon'/(\Delta\sigma^2/50) \leq \epsilon$.

For the sample complexity of a single eigendecomposition, we have to take enough samples so that for $8000 \log^3 n$ different instantiations of $D^2\psi(u)$, the spectral norm error is within ϵ' with high probability. It suffices to estimate three matrix-valued random variables $\mathbb{E}\left(xx^T \exp(iu^T x)\right)$, $\mathbb{E}\left(x \exp(iu^T x)\right)$ and $\mathbb{E}\left(\exp(iu^T x)\right)$. The latter two are easy to estimate using $O(n)$ samples by applying Lemma 9. Thus, it suffices for us to show that we can estimate the second order term $\mathbb{E}\left(xx^T \exp(iu^T x)\right)$ using only a nearly linear number of samples. We rewrite this term as four easy-to-estimate parts:

$$\begin{aligned}\mathbb{E}\left(xx^T \exp(iu^T x)\right) &= \mathbb{E}\left(xx^T \mathbb{1}_{\cos(u^T x)\geq 0} \cos(u^T x)\right) - \mathbb{E}\left(xx^T \mathbb{1}_{\cos(u^T x)<0} \left|\cos(u^T x)\right|\right) \\ &\quad + i\mathbb{E}\left(xx^T \mathbb{1}_{\sin(u^T x)\geq 0} \sin(u^T x)\right) - i\mathbb{E}\left(xx^T \mathbb{1}_{\sin(u^T x)<0} \left|\sin(u^T x)\right|\right)\end{aligned}$$

We estimate the four terms using independent samples to within error $\epsilon/4$ in spectral norm. Consider, for example, the first term $xx^T \mathbb{1}_{\cos u^T x \geq 0} \cos(u^T x)$, then we can define the random vector

$$y = x\mathbb{1}_{\cos(u^T x)\geq 0} \sqrt{\cos(u^T x)},$$

so that the first term is yy^T. In particular, observe that $0 \leq \mathbb{E}\left((u^T y)^2\right) \leq \mathbb{E}\left((u^T x)^2\right) \leq 1$ for all unit vectors u. Thus, we must have that the eigenvalues of $\mathbb{E}\left(yy^T\right)$ 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 Vershynin (2010) to y: by hypothesis, we can take $m = K^2 n$ and $t = 2$. Let N be the number of samples needed to get a failure probability smaller than $1/n^2$.

Next, we argue that we can re-use the N samples from the initial phase for the entire algorithm (without re-sampling), and apply the union bound to get a failure probability bounded by $1/n^2$, thereby giving us a high probability of success for the entire algorithm. To find a good vector u, we form the second derivative matrices for a set of u's, explicitly compute the eigenvalue gaps, find the largest gap for each matrix, and pick the u whose matrix has the largest gap. Finding a good u here depends only on the randomness of u. We note that each level of decomposition we estimate is of the form $\mathbb{E}\left((P^T x)(P^T x)^T \exp\left(iu^T x\right)\right) = P^T \mathbb{E}\left(xx^T \mathrm{diag}\left(\exp\left(iu^T x\right)\right)\right) P$, with the expectation independent of the current projection. We will do this for at most $O(n \log^3 n)$ different vectors u, which are random and independent of the sample. To achieve overall error ϵ, the sample complexity is thus

$$O\left(K^2 n \frac{\log n}{(\epsilon')^2}\right) = O\left(n \cdot \frac{K^2 \mathbf{M}^4 \log^7 n}{\Delta^6 \epsilon^2}\right).$$

■

5. Conclusion

Our work was motivated by experiments on Fourier PCA and tensor-based methods, which appeared to need a rather large number of samples even for modest values of the dimension n. In contrast, the recursive algorithm presented here scales smoothly with the dimension, and is available as MATLAB code (Xiao, 2014).

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 $O(1/n^{0.6})$, no longer polylogarithmic as in the case of degree 1 or 2.

References

Anima Anandkumar, Dean Foster, Daniel Hsu, Sham Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems 25*, pages 926–934, 2012a.

Anima Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *CoRR*, abs/1210.7559, 2012b.

Joseph Anderson, Mikhail Belkin, Navin Goyal, Luis Rademacher, and James Voss. The more, the merrier: the blessing of dimensionality for learning large gaussian mixtures. *arXiv:1311.2891*, 2013.

Sanjeev Arora, Rong Ge, Ankur Moitra, and Sushant Sachdeva. Provable ICA with unknown gaussian noise, with implications for gaussian mixtures and autoencoders. In *NIPS*, pages 2384–2392, 2012.

Mikhail Belkin, Luis Rademacher, and James Voss. Blind signal separation in the presence of Gaussian noise. In *Proc. of COLT*, 2013.

C. Brubaker. Extensions of principal component analysis. *Phd. Thesis, School of CS, Georgia Tech*, 2009.

J.F. Cardoso. Source separation using higher order moments. In *International Conference on Acoustics, Speech, and Signal Processing*, 1989.

Pierre Comon and Christian Jutten, editors. *Handbook of Blind Source Separation*. Academic Press, 2010.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Paul Deheuvels. Strong limit theorems for maximal spacings from a general univariate distribution. *The Annals of Probability*, pages 1181–1193, 1984.

Paul Deheuvels. The limiting behaviour of the maximal spacing generated by an iid sequence of gaussian random variables. *Journal of applied probability*, pages 816–827, 1985.

Paul Deheuvels. On the influence of the extremes of an iid sequence on the maximal spacings. *The Annals of Probability*, pages 194–208, 1986.

William Feller. *An Introduction to Probability Theory and its Applications, vol 1*. John Wiley and sons, 1968.

Alan M. Frieze, Mark Jerrum, and Ravi Kannan. Learning linear transformations. In *FOCS*, pages 359–368, 1996.

Navin Goyal, Santosh Vempala, and Ying Xiao. Fourier PCA and robust tensor decomposition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 584–593, 2014.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

Christopher Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60, 2013.

Daniel Hsu and Sham M. Kakade. Learning mixtures of spherical Gaussians: moment methods and spectral decompositions. In *ITCS*, pages 11–20, 2013.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley, 2001.

Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang W Koh, Quoc V Le, and Andrew Y Ng. Tiled convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1279–1287, 2010.

Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *J. Cryptology*, 22(2):139–160, 2009.

Gilbert W Stewart and Ji-guang Sun. *Matrix perturbation theory*. Academic press, 1990.

Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing, Theory and Applications*, pages 210–268. Cambridge University Press, Oxford, 2010.

Ying Xiao. Fourier pca package. *GitHuB*, 2014. http://github.com/yingusxiaous/libFPCA.

Arie Yeredor. Blind source separation via the second characteristic function. *Signal Processing*, 80 (5):897–902, 2000.

6. Appendix

Claim 11 *Let $a, b \in [0,1]$, $a \leq b^2/8$, and define the recurrence:*

$$y_0 = 0, \quad y_{i+1} = (1 + y_i^2)\left(a + \left(\frac{y_i}{b}\right)^2\right).$$

Then $y_i \leq 2a$ for all i.

Proof We proceed via induction. Clearly this is true for $i = 0$. Now suppose it is true for up to some i. Then

$$y_{i+1} = (1 + 4a^2)(a + (2a/b)^2) \leq a\left(1 + 4a^2 + 4\frac{a}{b^2} + 16\frac{a^3}{b^2}\right) \leq 2a$$

since $a \leq b^2/8 \leq 1/8$. ■

Acknowlegements. We are grateful to the anonymous referees for their suggestions. This work was supported in part by NSF awards CCF-1217793 and EAGER-1415498.