Noname manuscript No.
(will be inserted by the editor)

Provable ICA with Unknown Gaussian Noise, and Implications for Gaussian Mixtures and Autoencoders

Sanjeev Arora · Rong Ge · Ankur Moitra · Sushant Sachdeva

Received: date / Accepted: date

Abstract We present a new algorithm for Independent Component Analysis (ICA) which has provable performance guarantees. In particular, suppose we are given samples of the form $y = Ax + \eta$ where A is an unknown but non-singular $n \times n$ matrix, x is a random variable whose coordinates are independent and have a fourth order moment strictly less than that of a standard Gaussian random variable and η is an n-dimensional Gaussian random variable with unknown covariance Σ: We give an algorithm that provably recovers A and Σ up to an additive ϵ and whose running time and sample complexity are polynomial in n and $1/\epsilon$. To accomplish this, we introduce a novel "quasi-whitening" step that may be useful in other applications where there is additive Gaussian noise whose covariance is unknown. We also give a general framework for finding all local optima of a function (given an oracle for approximately finding just one) and this is a crucial step in our algorithm,

Sanjeev Arora
Princeton University
Princeton, NJ
E-mail: arora@cs.princeton.edu

Rong Ge
Microsoft Research
Cambridge, MA
E-mail: rongge@microsoft.com

Ankur Moitra
Massachusetts Institute of Technology
Cambridge, MA
E-mail: moitra@mit.edu

Sushant Sachdeva
Yale University
New Haven, CT
E-mail: sachdeva@cs.princeton.edu

one that has been overlooked in previous attempts, and allows us to control the accumulation of error when we find the columns of A one by one via local search.

Keywords independent component analysis · mixture models · method of moments · cumulants

1 Introduction

We present an algorithm (with rigorous performance guarantees) for a basic statistical problem. Suppose η is an independent n-dimensional Gaussian random variable with an unknown covariance matrix Σ and A is an unknown but non-singular $n \times n$ matrix. We are given samples of the form $y = Ax + \eta$ where x is a random variable whose coordinates are independent, mean zero and have a fourth order moment strictly less than that of a Gaussian random variable with same variance. The most natural case is when x is chosen uniformly at random from $\{+1, -1\}^n$, although our algorithms work in the more general case above. Our goal is to reconstruct an additive approximation to the matrix A and the covariance matrix Σ running in time and using a number of samples that is polynomial in n and $\frac{1}{\epsilon}$, where ϵ is the target precision (see Theorem 1). This problem arises in several applications within machine learning: Independent Component Analysis (ICA), Deep Learning, Gaussian Mixture Models (GMM), etc. We describe these connections next, and known results (focusing on algorithms with provable performance guarantees, since that is our goal).

Most obviously, the above problem can be seen as an instance of *Independent Component Analysis* (ICA) with unknown Gaussian noise. ICA has an illustrious history with applications ranging from econometrics, to signal processing, to image segmentation. The goal generally involves finding a linear transformation of the data so that the coordinates are as independent as possible [9] [22] [24]. This is often accomplished by finding directions in which the projection is "non Gaussian" [21]. Clearly, if the datapoint y is generated as Ax (i.e., with no noise η added) then applying linear transformation A^{-1} to the data results in samples $A^{-1}y$ whose coordinates are independent. This noiseless case was considered by Comon [9] and Frieze, Jerrum and Kannan [16], and their goal was to recover an additive approximation to A efficiently and using a polynomial number of samples. We will later note a gap in their reasoning, albeit fixable by our methods. To the best of our knowledge, prior to our work there were no known algorithms for ICA with Gaussian noise with provable guarantees. Here we require that our algorithms run in polynomial time, and that their estimates converge at an inverse polynomial rate to the true values as we increase the number of samples. See also concurrent and independent work by Anandkumar *et al.* [1], Hsu and Kakade [20], that give alternative, efficient algorithms that use tensor decompositions instead of local search, as we do here.

The second view of our problem is as a compactly described *Gaussian Mixture Model*. Our data is generated as a mixture of 2^n identical Gaussian coordinates (with an unknown covariance matrix) whose centers are the points $\{Ax : x \in \{-1,1\}^n\}$, and all mixing weights are equal. Notice, this mixture of 2^n Gaussians can be described using $O(n^2)$ parameters. The problem of learning Gaussian mixtures has a long history, and the popular approach in practice is to use the EM algorithm [14], though it has no worst-case guarantees (the method may take a very long time to converge, and worse, may not always converge to the correct solution). An influential paper of Dasgupta [11] initiated the program of designing algorithms with provable guarantees, which was improved in a sequence of papers [3], [5], [25], [27]. But in the current setting, it is unclear how to apply any of the above algorithms (including *EM*) since the trivial application would keep track of exponentially many parameters – one for each component. Thus, new ideas seem necessary to achieve polynomial running time.

The third view of our problem is as a simple form of *autoencoding* [19]. This is a central notion in Deep Learning, where the goal is to obtain a compact representation of a target distribution using a multilayered architecture, where a complicated function (the target) can be built up by composing layers of a simple function (called the autoencoder [6]). The main tenet is that there are interesting functions which can be represented concisely using many layers, but would need a very large representation if a "shallow" architecture is used instead. This is most useful for functions that are "highly varying" (i.e. cannot be compactly described by piecewise linear functions or other "simple" local representations). Formally, it is possible to represent using just (say) n^2 parameters, some distributions with 2^n "varying parts" or "interesting regions." The *Restricted Boltzmann Machine* (RBM) is an especially popular autoencoder in Deep Learning, though many others have been proposed. However, to the best of our knowledge, there has been no successful attempt to give a *rigorous* analysis of Deep Learning. Concretely, if the data is indeed generated using the distribution represented by an RBM, then do the popular algorithms for Deep Learning [18] learn the model parameters *correctly* and in *polynomial* time? Clearly, if the running time were actually found to be exponential in the number of parameters, then this would erode some of the advantages of the compact representation.

How is Deep Learning related to our problem? As noted by Freund and Haussler [15], an RBM with real-valued visible units (the version that seems more amenable to theoretical analysis) is precisely a mixture of exponentially many standard Gaussians. It is parametrized by an $n \times m$ matrix A and a vector $\theta \in \mathbb{R}^n$. It encodes a mixture of n-dimensional standard Gaussians centered at the points $\{Ax : x \in \{-1,1\}^m\}$, where the mixing weight of the Gaussian centered at Ax is $\exp(\|Ax\|_2^2 + \theta \cdot x)$. This is of course reminiscent of our problem. Formally, our algorithm can be seen as a nonlinear autoencoding scheme analogous to an RBM but with uniform mixing weights. Interestingly, the algorithm that we present here looks nothing like the approaches favored

traditionally in Deep Learning, and may provide an interesting new perspective.

1.1 Our Results and Techniques

We give a provable algorithm for ICA with unknown Gaussian noise. We have not made an attempt to optimize its running time, but we emphasize that this is in fact the first algorithm with provable guarantees for this problem and moreover we believe that in practice our algorithm will run almost as fast as the usual ICA algorithms, which are its close relatives.

Theorem 1 (Main, Informally, see Theorem 7) *There is an algorithm that recovers the unknown A and Σ up to additive error ϵ in each entry in time that is polynomial in $n, \|A\|_2, \|\Sigma\|_2, 1/\epsilon, 1/\lambda_{\min}(A)$ where $\|\cdot\|_2$ denotes the operator norm and $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue.*

The classical approach for ICA initiated in Comon [9] and Frieze, Jerrum and Kannan [16] works in the noiseless case where we have $y = Ax$. The first step is *whitening*, which applies a suitable linear transformation that makes the variance the same in all directions, thus reducing to the case where A is an *orthogonal* matrix. Given samples $y = Rx$ where R is an orthogonal matrix, the rows of R can be found in principle by computing the vectors u that are the exact local minima of $E[(u \cdot y)^4]$. Subsequently, a number of works (see e.g. [10] [13]) have focused on giving algorithms that are robust to noise. A popular approach is to use the fourth order *cumulant* (as an alternative to the fourth order moment) as a method for "denoising," or any one of a number of other functionals whose local optima reveal interesting directions. However, theoretical guarantees of these algorithms are not well understood. For example, many known approaches assume exact access to various statistics, but provide no sample complexity bounds in order to calculate these or no bounds on the error of their estimator when given noisy approximations to the statistics.

The above procedures in the noise-free model can *almost* be made rigorous (i.e., provably polynomial running time and number of samples), except for one subtlety: it is unclear how to use local search to find *all* optima in polynomial time. In practice, one finds a single local optimum, projects to the subspace orthogonal to it and continues recursively on a lower-dimensional problem. However, a naive implementation of this idea is unstable since approximation errors can accumulate badly, and to the best of our knowledge no rigorous analysis has been given prior to our work. (This is not a technicality: in some similar settings the errors are known to blow up exponentially [28].) One of our contributions is a simple framework for analyzing local search that avoids this issue, and is able to find all local optima without error accumulating badly. (Section 5.2.)

Our major new contribution however is dealing with noise that is an unknown Gaussian. This is an important generalization, since many methods

used in ICA are quite unstable to noise (and a wrong estimate for the covariance could lead to bad results). Here, we no longer need to assume we know even rough estimates for the covariance. Moreover, in the context of Gaussian Mixture Models this generalization corresponds to learning a mixture of identical but non-spherical Gaussians where the common covariance of the components is not known in advance.

We design new tools for denoising and especially whitening in this setting. Denoising uses the fourth order cumulant instead of the fourth order moment used in [16] and whitening involves a novel use of the Hessian of the cumulant. Even then, we cannot reduce to the simple case $y = Rx$ as above, and are left with a more complicated functional form (see "quasi-whitening" in Section 3.) Nevertheless, we can reduce to an optimization problem that can be solved via local search, and which remains amenable to a rigorous analysis and from which we can recover A as well as the covariance Σ of the noise.

In order to avoid cluttered notation, we have focused on the case in which x is chosen uniformly at random from $\{-1, +1\}^n$, although our algorithm and analysis work under the more general conditions that the coordinates of x are (i) independent with mean zero and unit variance; (ii) have a fourth order moment that is less than three (the fourth order moment of a Gaussian random variable). In this case, the functional $P(u)$ (see Lemma 1) will take the same form but with weights depending on the exact value of the fourth order moment for each coordinate. Since we already carry through an unknown diagonal matrix D throughout our analysis, this generalization only changes the entries on the diagonal and the same algorithm and proof apply.

Subsequent Work

There has been considerable recent work on Independent Component Analysis and related problems. Belkin, Rademacher and Voss [4] recently gave an improved algorithm for ICA with unknown Gaussian noise that works when each of the coordinates of x has a fourth order moment that is bounded away from three (instead of strictly less than three, as in our setting). Goyal, Vempala and Xiao [17] gave an algorithm for overcomplete ICA that works even when A is an $n \times m$ matrix and $m \gg n$ (see also [12], [29]). There have also been a number of recent works giving improved algorithms for learning mixtures of Gaussians where the number of components is a fixed polynomial in the dimension [17], [8], [2]. However these results are incomparable since in our setting the number of components is exponential but we require their centers to be the image of the hypercube under a linear transformation.

2 Overview

In this section we give an overview for the main ideas of our algorithm (see Algorithm 1, for more details see Section 4).

Algorithm 1. MainAlgorithm, **Input:**samples,ϵ **Output:** matrices $\widehat{A}$, $\widehat{\Sigma}$ with accuracy ϵ

(**Denoising and Quasi-whitening**)

1. Pick a random $u_0 \sim \mathcal{N}(0, \frac{1}{n} I_n)$
2. Take $2N$ samples $y_1, y_2, ..., y_N, y'_1, y'_2, ..., y'_N$ ($N = \text{poly}(n\|A\|_2, \|\Sigma\|_2, 1/\epsilon, 1/\lambda_{min}(A))$), estimate the Hessian $\mathcal{H}(P(u_0))$ of the 4-th order cumulant $P(u_0) = -\kappa_4(u^T y) = 2\sum_{i=1}^n (u^T A)_i^4$ (see Section 3 for more details).

$$\mathcal{H}(\widehat{P}(u_0)) = \frac{1}{N}\sum_{i=1}^{N} 12(u_0^T y_i)^2 y_i y_i^T + 2(u_0^T y'_i)^2 y_i y_i^T + 2(u_0^T y_i)^2 y'_i (y'_i)^T + 4(u_0^T y_i)(u_0^T y'_i)(y_i (y'_i)^T + (y'_i) y_i^T).$$

3. Compute B such that $\mathcal{H}(\widehat{P}(u_0)) = BB^T$.

(**Finding Local Maxima**)

4. Estimate

$$\widehat{P}'(u) = -\frac{1}{N}\sum_{i=1}^{N} (u^T B^{-1} y_i)^4 + \frac{3}{N}\left(\sum_{i=1}^{N} (u^T B^{-1} y_i)^2 (u^T B^{-1} y'_i)^2\right)$$

which is an empirical estimation of $P'(u)$ (which is the 4-th order cumulant in the whitened coordinate system, later in Section 3 defined to be $P'(u) = \kappa_4(u^T z) = \kappa_4(u^T B^{-1} y)$).

5. Choose parameters β and δ to be $\text{poly}(n\|A\|_2, \|\Sigma\|_2, 1/\epsilon, 1/\lambda_{min}(A))$ according to Theorem 10.
6. Use Algorithm 3 AllOpt$(\widehat{P}'(u), \beta, \delta', \beta', \delta')$ of Section 5 to compute all n local maxima of the function $\widehat{P}'(u)$.

(**Recovering Parameters**)

7. Let R be the matrix whose rows are the n local optima recovered in the previous step.
8. Use Algorithm 4 Recover of Section 6 to find A and Σ

Before explaining our algorithm, let us first recall previous approaches. The approach of Frieze Jerrum and Kannan [16] is to approximate the second moment matrix. One can then use this to find a linear transformation so that applying it transforms the problem to an instance where the columns of A are almost orthogonal. This is often called whitening. After this transformation, the second step of the algorithm attempts to maximize a function of the 4th order moments and the analysis proceeds by showing that this function has exactly $2n$ distinct local maxima that correspond to the columns of the (transformed) A matrix (with sign flips). Finally in the last step one can recover the original A by undoing the linear transformation.

Our algorithm has three similar steps: Denoising and Quasi-whitening, Finding Local Maxima and Recovering Parameters.

Denoising and Quasi-whitening In our setting the model has Gaussian noise, so the moment structure is very different from the noiseless case. We first observe that if we use 4th order cumulants, instead of 4th order moments, our

statistics will be unaffected by the Gaussian noise. However, the second order cumulants and moments are the same, and they both depend on the Gaussian noise. In order to perform (quasi-)whitening, we observe that the singular value decomposition of a certain Hessian matrix (which we can estimate from samples) can used to find a suitable linear transformation. We can think of this step as a replacement for the one in [16] which is affected by Gaussian noise. See Section 3 for details.

Finding the Parameters After quasi-whitening, the problem essentially reduces to the following:

Question: Suppose there is a function f which has exactly n orthogonal local maxima that correspond to the columns of A. Given an empirical estimate $\hat{f}$ of this function, can we find vectors that are close to the local maxima of f?

Frieze Jerrum and Kannan[16] showed how to find one local maximum. However it was not clear how this could generalize to finding all the local maxima as naively the errors accumulate exponentially in terms of the dimension. We identify properties of the function f (namely locally approximable and locally improvable, see Section 5), and show that in general under these assumptions it is possible to find all local maxima using a delicate two-step local search procedure. See Section 5 for details.

Recovering the Parameters This is a standard step, we use basic linear algebra to invert the (quasi-)whitening transformation and estimate the covariance Σ. We also prove polynomial bounds on the sample complexity. See Theorem 12.

3 Denoising and Quasi-Whitening

As mentioned, our approach is based on the fourth order cumulant. The cumulants of a random variable are the coefficients of the Taylor expansion of the logarithm of the characteristic function [26]. Let $\kappa_r(X)$ be the r^{th} cumulant of a random variable X. We make use of:

Fact 2 *(i) If X has mean zero, then $\kappa_4(X) = \mathbb{E}[X^4] - 3\mathbb{E}[X^2]^2$. (ii) If X is Gaussian with mean μ and variance σ^2, then $\kappa_1(X) = \mu$, $\kappa_2(X) = \sigma^2$ and $\kappa_r(X) = 0$ for all $r > 2$. (iii) If X and Y are independent, then $\kappa_r(X + Y) = \kappa_r(X) + \kappa_r(Y)$.*

The crux of our technique is to look at the following functional, where y is the random variable $Ax + \eta$ whose samples are given to us. Let $u \in \mathbb{R}^n$ be any vector. Then $P(u) = -\kappa_4(u^T y)$. Note that for any u we can compute $P(u)$ reasonably accurately by drawing sufficient number of samples of y and taking an empirical average. Furthermore, since x and η are independent, and η is Gaussian, the next lemma is immediate. We call it "denoising" since it allows us empirical access to some information about A that is uncorrupted by the noise η.

Lemma 1 (Denoising Lemma) $P(u) = 2\sum_{i=1}^{n}(u^T A)_i^4$.

Proof The crucial observation is that $u^T y = u^T Ax + u^T \eta$ is the sum of two independent random variables, Ax and η and that $P(u) = -\kappa_4(u^T Ax + u^T \eta) = -\kappa_4(u^T Ax) - \kappa_4(u^T \eta) = -\kappa_4(u^T Ax)$. So in fact, the functional $P(u)$ is invariant under additive Gaussian noise **independent of the variance matrix** Σ. This vastly simplifies our computation:

$$\begin{aligned}\mathbb{E}[(u^T Ax)^4] &= \sum_{i=1}^{n}(u^T A)_i^4 \mathbb{E}[x_i^4] + 6\sum_{i<j}(u^T A)_i^2 (u^T A)_j^2 \mathbb{E}[x_i^2]\mathbb{E}[x_j^2] \\ &= \sum_{i=1}^{n}(u^T A)_i^4 + 6\sum_{i<j}(u^T A)_i^2 (u^T A)_j^2 = -2\sum_{i=1}^{n}(u^T A)_i^4 + 3(u^T AA^T u)^2\end{aligned}$$

Furthermore $\mathbb{E}[(u^T Ax)^2]^2 = (u^T AA^T u)^2$ and we conclude that

$$P(u) = -\kappa_4(u^T y) = -\mathbb{E}[(u^T Ax)^4] + 3\mathbb{E}[(u^T Ax)^2]^2 = 2\sum_{i=1}^{n}(u^T A)_i^4.$$

3.1 Quasi-Whitening via the Hessian of $P(u)$

In prior works on ICA, *whitening* refers to reducing to the case where $y = Rx$ for some some orthogonal matrix R. Here we give a technique to reduce to the case where $y = RDx + \eta'$ where η' is some other Gaussian noise (still unknown), R is an orthogonal matrix and D is a diagonal matrix that depends upon A. We call this *quasi-whitening*. Quasi-whitening suffices for us since local search using the objective function $\kappa_4(u^T y)$ will give us (approximations to) the rows of RD, from which we will be able to recover A.

Quasi-whitening involves computing the Hessian of $P(u)$, which recall is the matrix of all 2nd order partial derivatives of $P(u)$. Throughout this section, we will denote the Hessian operator by $\mathcal{H}$. In matrix form, $\mathcal{H}P(u)$ is

$$\begin{aligned}\mathcal{H}P(u) = \frac{\partial^2}{\partial u_i \partial u_j}P(u) &= 24\sum_{k=1}^{n} A_{i,k}A_{j,k}(A_k \cdot u)^2 \\ &= 24\sum_{k=1}^{n}(A_k \cdot u)^2 A_k A_k^T = AD_A(u)A^T\end{aligned}$$

where A_k is the k-th column of the matrix A (we use subscripts to denote the columns of matrices throught the paper). $D_A(u)$ is the following diagonal matrix:

Definition 1 Let $D_A(u)$ be a diagonal matrix in which the k^{th} entry is $24(A_k \cdot u)^2$.

Of course, the exact Hessian of $P(u)$ is unavailable and we will instead compute an empirical approximation $\widehat{P}(u)$ to $P(u)$ (given many samples from the distribution), and we will show that the Hessian of $\widehat{P}(u)$ is a good approximation to the Hessian of $P(u)$.

Definition 2 Given $2N$ samples $y_1, y_1', y_2, y_2' ..., y_N, y_N'$ of the random variable y, let

$$\widehat{P}(u) = \frac{-1}{N}\sum_{i=1}^{N}(u^T y_i)^4 + \frac{3}{N}\sum_{i=1}^{N}(u^T y_i)^2(u^T y_i')^2.$$

Our first step is to show that the expectation of the Hessian of $\widehat{P}(u)$ is exactly the Hessian of $P(u)$. In fact, since the expectation of $\widehat{P}(u)$ is exactly $P(u)$ (and since $\widehat{P}(u)$ is an analytic function of the samples and of the vector u), we can interchange the Hessian operator and the expectation operator. Roughly, one can imagine the expectation operator as an integral over the possible values of the random samples, and as is well-known in analysis, one can differentiate under the integral provided that all functions are suitably smooth over the domain of integration.

Claim 3 $\mathbb{E}_{y,y'}[-(u^T y)^4 + 3(u^T y)^2(u^T y')^2] = P(u)$

This claim follows immediately from the definition of $P(u)$, and since y and y' are independent.

Lemma 2 $\mathcal{H}(P(u)) = \mathbb{E}_{y,y'}[\mathcal{H}(-(u^T y)^4 + 3(u^T y)^2(u^T y')^2)]$

Next, we compute the two terms inside the expectation:

Claim 4 $\mathcal{H}((u^T y)^4) = 12(u^T y)^2 yy^T$

Claim 5 $\mathcal{H}((u^T y)^2(u^T y')^2) = 2(u^T y')^2 yy^T + 2(u^T y)^2 y'(y')^T + 4(u^T y)(u^T y')(y(y')^T + (y')y^T)$

Let $\lambda_{\min}(A)$ denote the smallest eigenvalue of A. Our analysis also requires bounds on the entries of $D_A(u_0)$:

Claim 6 *If u_0 is a random Gaussian variable $\mathcal{N}(0, \frac{1}{n}I_n)$ with expected square norm 1, then with probability $1 - O(1/\sqrt{n})$ we have for all i,*

$$\min_{i=1}^{n} \|A_i\|_2^2 n^{-4} \leq D_A(u_0)_{i,i} \leq \max_{i=1}^{n} \|A_i\|_2^2 \frac{9\log n}{n}.$$

Proof Since u_0 is a Gaussian random variable, each of the $A_i \cdot u$ is distributed as a Gaussian with mean zero and variance $\|A_i\|^2/n$.

By the tail properties of Gaussian distribution, we know $\Pr[|A_i \dot{u}_0| \geq \|A_i\|\frac{3\sqrt{\log n}}{n}] \leq n^{-2}$. On the other hand, by the anticoncentration properties of Gaussians, $\Pr[|A_i \dot{u}_0| \leq \|A_i\| n^{-2}] \leq O(n^{-1.5})$. Hence by union bound with probability at least $1 - O(1/\sqrt{n})$, none of the events $|A_i \dot{u}_0| \geq \|A_i\|\frac{3\sqrt{\log n}}{n}$, $|A_i \dot{u}_0| \leq \|A_i\| n^{-2}$ happen. In this case we know $(D_A(u_0))_{i,i} = |A_i \cdot u_0|^2$ is in between $\min_{i=1}^n \|A_i\|_2^2 n^{-4}$ and $\max_{i=1}^n \|A_i\|_2^2 \frac{9\log n}{n}$.

Lemma 3 *If u_0 is chosen randomly from $\mathcal{N}(0, \frac{1}{n}I_n)$ and furthermore we are given $2N = \text{poly}(n, 1/\epsilon, 1/\lambda_{\min}(A), \|A\|_2, \|\Sigma\|_2)$ samples of y, then with probability $1 - O(1/\sqrt{n})$ we will have that $(1-\epsilon)AD_A(u_0)A^T \preceq \mathcal{H}(\widehat{P}(u_0)) \preceq (1+\epsilon)AD_A(u_0)A^T$.*

Proof The Hessian of $\widehat{P}(u_0)$ is the sum of independent random matrices (see Definition 2. First we consider entry-wise bounds for matrices in this sum. For example, the variance of any entry in $\mathcal{H}((u^Ty)^4) = 12(u^Ty)^2yy^T$ can be bounded by $O(\|y\|_2^8)$ (with probability $1 - \exp(-\Omega(n))$), which we can then bound by $\mathbb{E}[\|y\|_2^8] \leq O(\mathbb{E}[\|Ax\|_2^8 + \|\eta\|_2^8])$. This can be bounded by $O(n^4(\|A\|_2^8 + \|\Sigma\|_2^4))$. This is also an upper bound for the variance (of any entry) when computing $\mathcal{H}(\widehat{P}(u_0))$ (the other terms have smaller variance).

Applying standard concentration bounds, $\text{poly}(n, 1/\epsilon', \|A\|_2, \|\Sigma\|_2)$ samples suffice to guarantee that all entries of $\mathcal{H}(\widehat{P}(u_0))$ are ϵ' close to $\mathcal{H}(P(u))$. The smallest eigenvalue of $\mathcal{H}(P(u)) = AD_A(u_0)A^T$ is at least

$$\lambda_{\min}(A)^2 \min_{i=1}^{n} \|A_i\|_2^2 n^{-4}$$

where here we have used Claim 6. If we choose $\epsilon' = \text{poly}(1/n, \lambda_{\min}(A), \epsilon)$, then we are also guaranteed $(1-\epsilon)AD_A(u_0)A^T \preceq \mathcal{H}(\widehat{P}(u_0)) \preceq (1+\epsilon)AD_A(u_0)A^T$ holds.

Lemma 4 *Suppose that $(1-\epsilon)AD_A(u_0)A^T \preceq \widehat{M} \preceq (1+\epsilon)AD_A(u_0)A^T$, and let $\widehat{M} = BB^T$. Then there is an orthogonal matrix R^* such that $\|B^{-1}AD_A(u_0)^{1/2} - R^*\|_F \leq \sqrt{n}\epsilon$.*

The intuition is: if any of the singular values of $B^{-1}AD_A(u_0)^{1/2}$ are outside the range $[1-\epsilon, 1+\epsilon]$, we can find a unit vector x where the quadratic forms $x^TAD_A(u_0)A^Tx$ and $x^T\widehat{M}x$ are too far apart (which contradicts the condition of the lemma). Hence the singular values of $B^{-1}AD_A(u_0)^{1/2}$ can all be set to one without changing the Frobenius norm of $B^{-1}AD_A(u_0)^{1/2}$ too much, and this yields an orthogonal matrix.

Proof Let $M = AD_A(u_0)A^T$ and let $C = AD_A(u_0)^{1/2}$, and so $M = CC^T$ and $\widehat{M} = BB^T$. The condition $(1-\epsilon)M \preceq \widehat{M} \preceq (1+\epsilon)M$ is well-known to be equivalent to the condition that for all vectors x, $(1-\epsilon)x^TMx \leq x^T\widehat{M}x \leq (1+\epsilon)x^TMx$.

Suppose for the sake of contradiction that $S = B^{-1}C$ has a singular value outside the range $[1-\epsilon, 1+\epsilon]$. Assume (without loss of generality) that S has a singular value strictly larger than $1+\epsilon$ (and the complementary case can be handled analogously). Hence there is a unit vector y such that $y^TSS^Ty > 1+\epsilon$. But since $BSS^TB^T = CC^T$, if we set $x^T = y^TB^{-1}$ then we have $x^T\widehat{M}x = x^TBB^Tx = y^Ty = 1$ but $x^TMx = x^TCC^Tx = x^TBSS^TB^Tx = y^TSS^Ty > 1+\epsilon$. This is a contradiction and so we conclude that all of the singular values of $B^{-1}C$ are in the range $[1-\epsilon, 1+\epsilon]$.

Let $U\Sigma V^T$ be the singular value decomposition of $B^{-1}C$. If we set all of the diagonal entries in Σ to 1 we obtain an orthogonal matrix $R^* = UV^T$. And since the singular values of $B^{-1}C$ are all in the range $[1-\epsilon, 1+\epsilon]$, we can bound the Frobenius norm of $B^{-1}C - R^*$: $\|B^{-1}C - R^*\|_F \leq \sqrt{n}\epsilon$, as desired.

4 Our Algorithm (and Notation)

In this section we describe our overall algorithm. It uses as a blackbox the denoising and quasi-whitening already described above, as well as a routine for computing all local maxima of some "well-behaved" functions which is described later in Section 5.

Notation: Placing a hat over a function corresponds to an empirical approximation that we obtain from random samples. This approximation introduces error, which we will keep track of.

Step 1: *Pick a random* $u_0 \sim \mathcal{N}(0, \frac{1}{n}I_n)$ *and estimate the Hessian* $\mathcal{H}(\widehat{P}(u_0))$. *Compute* B *such that* $\mathcal{H}(\widehat{P}(u_0)) = BB^T$.

Step 2: *Take* $2N$ *samples* $y_1, y_2, ..., y_N, y'_1, y'_2, ..., y'_N$, *and let*

$$\widehat{P}'(u) = -\frac{1}{N}\sum_{i=1}^{N}(u^T B^{-1} y_i)^4 + \frac{3}{N}\left(\sum_{i=1}^{N}(u^T B^{-1} y_i)^2 (u^T B^{-1} y'_i)^2\right)$$

which is an empirical estimation of $P'(u)$ *(later defined to be* $P'(u) = \kappa_4(u^T z) = \kappa_4(u^T B^{-1} y)$*).*

Step 3: *Use the procedure* ALLOPT$(\widehat{P}'(u), \beta, \delta', \beta', \delta')$ *of Section 5 to compute all* n *local maxima of the function* $\widehat{P}'(u)$ *(since* $\widehat{P}'(u)$ *is symmetric* u *and* $-u$ *are considered as the same local maxima).*

Step 4: *Let* R *be the matrix whose rows are the* n *local optima recovered in the previous step. Use procedure* RECOVER *of Section 6 to find* A *and* Σ.

Explanation: Step 1 uses the transformation B^{-1} computed in the previous Section to quasi-whiten the data. Let $D = D_A(u_0)$ be the diagonal matrix defined in Definition 1. We consider the sequence of samples $z = B^{-1}y$, which are therefore of the form $R'D^{-1/2}x + \eta'$ where $\eta' = B^{-1}\eta$, $D = D_A(u_0)$ and R' is close to an orthogonal matrix R^* (by Lemma 4). In Step 2 we look at $\kappa_4((u^T z))$, which effectively denoises the new samples (see Lemma 1), and thus is the same as $\kappa_4(R'D^{-1/2}x)$. Let $P'(u) = \kappa_4(u^T z) = \kappa_4(u^T B^{-1} y)$ which is easily seen to be $E[(u^T R' D^{-1/2} x)^4]$. Step 2 estimates this function, obtaining $\widehat{P}'(u)$. Then Step 3 tries to find local optima via local search. Ideally we would have liked access to the functional $P^*(u) = (u^T R^* x)^4$ since the procedure for local optima works only for true orthogonal transformations. But since R' and R^* are close we can make it work approximately with $\widehat{P}'(u)$, and then in Step 4 use these local optima to finally recover A.

Theorem 7 *Suppose we are given samples of the form $y = Ax + \eta$ where x is uniform on $\{+1, -1\}^n$, A is an $n \times n$ matrix, η is an n-dimensional Gaussian random variable independent of x with unknown covariance matrix Σ. There is an algorithm that with high probability recovers $\|\widehat{A} - A\Pi Diag(s_i)\|_F \leq \epsilon$ where Π is some permutation matrix and each $s_i \in \{+1, -1\}$ and also recovers $\|\widehat{\Sigma} - \Sigma\|_F \leq \epsilon$. Furthermore the running time and number of samples needed are* $\mathrm{poly}(n, 1/\epsilon, \|A\|_2, \|\Sigma\|_2, 1/\lambda_{\min}(A))$

Proof In Step 1, by Lemma 4 we know once we use $z = B^{-1}y$, the whitened function $P'(u)$ is inverse polynomially close to $P^*(u)$. Then by Lemma 7, the function $\widehat{P'}(u)$ we get in Step 2 is inverse polynomially close to $P'(u)$ and $P^*(u)$. Theorem 9 and Lemma 9 show that given $\widehat{P'}(u)$ inverse polynomially close to $P^*(u)$, Algorithm 3: : AllOPT finds all local maxima with inverse polynomial precision. Finally by Theorem 12 we know A and W are recovered correctly up to additive ϵ error in Frobenius norm. The running time and sampling complexity of the algorithm is polynomial because all parameters in these Lemmas are polynomially related.

Note that here we recover A up to a permutation of the columns and sign-flips. In general, this is all we can hope for since the distribution of x is also invariant under these same operations. Also, the dependence of our algorithm on the various norms (of A and Σ) seems inherent since our goal is to recover an additive approximation, and as we scale up A and/or Σ, this goal becomes a stronger relative guarantee on the error.

5 Framework for Iteratively Finding all Local Maxima

In this section, we first describe a fairly standard procedure (based upon Newton's method) for finding a *single* local maximum of a function $f^* : \mathbb{R}^n \to \mathbb{R}$ among all unit vectors and an analysis of its rate of convergence. Such a procedure is a common tool in statistical algorithms, but here we state it rather carefully since we later give a general method to convert any local search algorithm (that meets certain criteria) into one that finds *all* local maxima (see Section 5.2).

Given that we can only ever hope for an additive approximation to a local maximum, one should be concerned about how the error accumulates when our goal is to find *all* local maxima. In fact, a naive strategy is to project onto the subspace orthogonal to the directions found so far, and continue in this subspace. However, such an approach seems to accumulate errors badly (the additive error of the last local maxima found is exponentially larger than the error of the first). Rather, the crux of our analysis is a novel method for bounding how much the error can accumulate (by refining old estimates).

Our strategy is to first find a local maximum in the orthogonal subspace, then run the local optimization algorithm again (in the original n-dimensional space) to "refine" the local maximum we have found. The intuition is that

since we are already close to a particular local maximum, the local search algorithm cannot jump to some other local maximum (since this would entail going through a valley).

5.1 Finding one Local Maximum

Throughout this section, we will assume that we are given oracle access to a function $f(u)$ and its gradient and Hessian. The procedure is also given a starting point u_s, a search range β, and a step size δ. For simplicity in notation we define the following projection operator.

Definition 3 $\text{Proj}_{\perp u}(v) = v - (u^T v)u$, $\text{Proj}_{\perp u}(M) = M - (u^T M u)uu^T$.

The basic step of the algorithm is a modification of Newton's method to find a local improvement that makes progress so long as the current point u is far from a local maximum. Notice that if we add a small vector to u, we do not necessarily preserve the norm of u. In order to have control over how the norm of u changes, during local optimization step the algorithm projects the gradient ∇f and Hessian $\mathcal{H}(f)$ to the space perpendicular to u. There is also an additional correction term $-\partial/\partial_u f(u) \cdot \|\xi\|^2/2$ (where ∂/∂_u is the directional derivative along direction u). This correction term is necessary because the new vector we obtain is $(u+\xi)/\left\|(u+\xi)\right\|_2$ which is close to $u - \|\xi\|_2^2/2 \cdot u + \xi + O(\beta^3)$. Therefore by Taylor's expansion, we have $f((u+\xi)/\left\|(u+\xi)\right\|_2) = \text{Proj}_{\perp u}(\nabla f(u))^T \xi + \frac{1}{2}\xi^T \text{Proj}_{\perp u}(\mathcal{H}(f(u)))\xi - \frac{1}{2}\left(\frac{\partial}{\partial_u} f(u)\right) \cdot \|\xi\|_2^2 + O(\beta^3)$. Step 2 of the algorithm is just maximizing a quadratic function and can be solved exactly (see Remark 1). To increase efficiency it is also acceptable to perform an approximate maximization step by taking ξ to be either aligned with the gradient $\text{Proj}_{\perp u}\nabla f(u)$ or the largest eigenvector of $\text{Proj}_{\perp u}(\mathcal{H}(f(u)))$.

Remark 1 In order to solve the optimization problem $\max_u \frac{1}{2}u^T A u - v^T u$ subject to $\|u\| \leq \beta$, observe that if the solution is inside the ball then u must be equal to $A^\dagger v$, we can first check whether this case is an optimal solution. If this is not optimal, then the optimal solution must be at the boundary and we have $(Au - v) = \lambda u$ for some $\lambda > 0$, further by second order conditions we have $\lambda > \lambda_{\max}(A)$. In this region the norm of $(A - \lambda I)^{-1} v$ is monotonically decreasing, so there is a unique value λ where $u = (A - \lambda I)^{-1} v$ have norm exactly β. This λ (and the corresponding solution u) can then be found using binary search.

The algorithm is guaranteed to succeed in polynomial time when the function is *Locally Improvable* and *Locally Approximable*:

Definition 4 ((γ, β, δ)-Locally Improvable) A function $f(u) : \mathbb{R}^n \to \mathbb{R}$ is (γ, β, δ)-Locally Improvable, if for any u that is at least γ far from any local maxima, there is a u' such that $\|u' - u\|_2 \leq \beta$ and $f(u') \geq f(u) + \delta$.

Algorithm 2. LocalOPT, **Input:** $f(u)$, u_s, β, δ **Output:** vector v

1. Set $u \leftarrow u_s$.
2. Maximize (see Remark 1): $\text{Proj}_{\perp u}(\nabla f(u))^T \xi + \frac{1}{2}\xi^T \text{Proj}_{\perp u}(\mathcal{H}(f(u)))\xi - \frac{1}{2}\left(\frac{\partial}{\partial_u} f(u)\right) \cdot \|\xi\|_2^2$
   Subject to $\|\xi\|_2 \leq \beta$ and $u^T \xi = 0$
3. Let ξ be the solution, $\tilde{u} = \frac{u+\xi}{\|u+\xi\|}$
4. If $f(\tilde{u}) \geq f(u) + \delta/2$, set $u \leftarrow \tilde{u}$ and Repeat Step 2
5. Else return u

Definition 5 ((β, δ)-Locally Approximable) A function $f(u)$ is locally approximable, if its third order derivatives exist and for any u and any direction v, the third order derivative of f at point u in the direction of v is bounded by $0.01\delta/\beta^3$.

Note that the definition of Locally Approximable only depends on the parameter δ/β^3. We choose to define it with two parameters (β, δ) so that its syntax matches the definition of (γ, β, δ)-Locally Improvable.

The analysis of the running time of the procedure comes from local Taylor expansion. When a function is Locally Approximable it is well approximated by the gradient and Hessian within a β neighborhood. The following theorem from [16] showed that the two properties above are enough to guarantee the success of a local search algorithm even when the function is only approximated.

Theorem 8 ([16], Lemma 11) *Let f, f^* be functions $\mathbb{R}^n \to \mathbb{R}$ whose 3rd order derivatives exists, if $|f(u) - f^*(u)| \leq \delta/8$ for all $u \in \mathbb{S}^{n-1}$, the function $f^*(u)$ is (γ, β, δ)-Locally Improvable, $f(u)$ is (β, δ) Locally Approximable, then Algorithm 2 will find a vector v that is γ close to some local maximum. The running time is at most $O((n^2 + T)\max f^*/\delta)$ where T is the time to evaluate the function f and its gradient and Hessian, and* $\max f^* = \max_{u \in S^{n-1}} f^*(u)$.

5.2 Finding all Local Maxima

Now we consider how to find *all* local maxima of a given function $f^*(u)$. The crucial condition that we need is that *all local maxima are orthogonal* (which is indeed true in our problem, and is morally true when using local search more generally in ICA). Note that this condition implies that there are at most n local maxima.[1] In fact we will assume that there are exactly n local maxima. If we are given an exact oracle for f^* and can compute *exact* local maxima then we can find all local maxima easily: find one local maximum, project the function into the orthogonal subspace, and continue to find more local maxima.

[1] Technically, there are $2n$ local maxima since for each direction u that is a local maxima, so too is $-u$ but this is an unimportant detail for our purposes.

Algorithm 3. AllOPT, **Input:** $f(u)$, β, δ, β', δ' **Output:** $v_1, v_2, ..., v_n$, $\forall i$ $\|v_i - v_i^*\| \leq \gamma$.

1. Let $v_1 = \text{LocalOPT}(f, e_1, \beta, \delta)$
2. FOR $i = 2$ TO n DO
3. Let g_i be the projection of f to the orthogonal subspace of $v_1, v_2, ..., v_{i-1}$.
4. Let $u' = \text{LocalOPT}(g, e_1, \beta', \delta')$.
5. Let $v_i = \text{LocalOPT}(f, u', \beta, \delta)$.
6. END FOR
7. Return $v_1, v_2, ..., v_n$

Definition 6 The projection of a function f to a linear subspace S is a function on that subspace with value equal to f. More explicitly, if $\{v_1, v_2, ..., v_d\}$ is an orthonormal basis of S, the projection of f to S is a function $g : \mathbb{R}^d \to \mathbb{R}$ such that $g(w) = f(\sum_{i=1}^{d} w_i v_i)$.

The following theorem gives sufficient conditions under which the above algorithm finds all local maxima, making precise the intuition given at the beginning of this section.

Theorem 9 *Suppose the function $f^*(u) : \mathbb{R}^n \to \mathbb{R}$ satisfies the following properties:*

(a) ***Orthogonal Local Maxima:*** *The function has n local maxima v_i^*, and they are orthogonal to each other.*

(b) ***Locally Improvable:*** *f^* is (γ, β, δ) Locally Improvable.*

(c) ***Improvable Projection:*** *The projection of the function to any subspace spanned by a subset of local maxima is $(\gamma', \beta', \delta')$ Locally Improvable. The step size $\delta' \geq 10\delta$.*

(d) ***Lipschitz:*** *If two points $\|u - u'\|_2 \leq 3\sqrt{n}\gamma$, then the function value $|f^*(u) - f^*(u')| \leq \delta'/20$.*

(e) ***Attraction Radius:*** *Let $Rad \geq 3\sqrt{n}\gamma + \gamma'$, for any local maximum v_i^*, let T be $\min f^*(u)$ for $\|u - v_i^*\|_2 \leq Rad$, then there exist a set U containing $\|u - v_i^*\|_2 \leq 3\sqrt{n}\gamma + \gamma'$ and does not contain any other local maxima, such that for every u that is not in U but is β close to U, $f^*(u) < T$.*

If we are given function f such that $|f(u) - f^*(u)| \leq \delta/8$ and f is both (β, δ) and $(\beta', \delta')-$Locally Approximable, then Algorithm 3 can find all local maxima of f^* within distance γ.

To prove this theorem, we first notice the projection of the function f in Step 3 of the algorithm should be close to the projection of f^* to the remaining local maxima. This is implied by Lipschitz condition and is formally shown in the following two lemmas. First we prove a "coupling" between the orthogonal complement of two close subspaces:

Lemma 5 *Given v_1, v_2, ..., v_k, each γ-close respectively to local maxima $v_1^*, v_2^*, ..., v_k^*$ (this is without loss of generality because we can permute the index of local maxima), then there is an orthonormal basis $v_{k+1}, v_{k+2}, ..., v_n$ for the*

orthogonal space of span$\{v_1, v_2, ..., v_k\}$ *such that for any unit vector* $w \in \mathbb{R}^{n-k}$, $\sum_{i=1}^{n-k} w_k v_{k+i}$ *is* $3\sqrt{n}\gamma$ *close to* $\sum_{i=1}^{n-k} w_k v^*_{k+i}$.

Proof Let S_1 be span$\{v_1, v_2, ..., v_k\}$, S_2 be span$\{v^*_1, v^*_2, ..., v^*_k\}$ and $S_1^\perp$, $S_2^\perp$ be their orthogonal subspaces respectively. We first prove that for any unit vector $v \in S_1^\perp$, there is another unit vector $v' \in S_2^\perp$ so that $v^T v' \geq 1 - 4n\gamma^2$. In fact, we can take v' to be the unit vector along the projection of v in $S_2^\perp$. To bound the length of the projection, we instead bound the length of projection to S_2. Since we know $v_i^T v = 0$ for $i \leq k$ and $\|v_i - v_i^*\| \leq \gamma$, it must be that $(v_i^*)^T v \leq 2\gamma$ when $\gamma < 0.01$. So the projection of v in S_2 has length at most $2\sqrt{n}\gamma$ and hence the projection of v in $S_2^\perp$ has length at least $1 - 4n\gamma^2$.

Next, we prove that there is a pair of orthornormal bases $\{\tilde{v}_{k+1}, \tilde{v}_{k+2}, ..., \tilde{v}_n\}$ and $\{\tilde{v^*}_{k+1}, \tilde{v^*}_{k+2}, ..., \tilde{v^*}_n\}$ for $S_1^\perp$ and $S_2^\perp$ such that $\sum_{i=1}^{n-k} w_k \tilde{v}_{k+i}$ is close to $\sum_{i=1}^{n-k} w_k \tilde{v^*}_{k+i}$. Once we have such a pair, we can simultaneously rotate the two bases so that the latter becomes $v^*_{k+1}, ..., v^*_n$.

To get this set of bases we consider the projection operator to $S_2^\perp$ for vectors in $S_1^\perp$. The squared length of the projection is a quadratic form over the vectors in $S_1^\perp$. So there is a symmetric PSD matrix M such that

$$\|\text{Proj}_{S_2^\perp}(v)\|_2^2 = v^T M v$$

for $v \in S_1^\perp$. Let $\{\tilde{v}_{k+1}, \tilde{v}_{k+2}, ..., \tilde{v}_n\}$ be the eigenvectors of this matrix M. As we showed the eigenvalues must be at least $1 - 8n\gamma^2$. The basis for $S_2^\perp$ will just be unit vectors along directions of projections of $\tilde{v}_i$ to $S_2^\perp$. They must also be orthogonal because the projection operator $\text{Proj}_{S_2^\perp}$ is linear and

$$\|Proj_{S_2^\perp}(\sum_{i=1}^{n-k} w_i \tilde{v}_{k+i})\|_2^2 = \|\sum_{i=1}^{n-k} w_i \text{Proj}_{S_2^\perp}(\tilde{v}_{k+i})\|_2^2 = \sum_{i=1}^{n-k} \lambda_i w_i^2$$

The second equality cannot hold if these vectors are not orthogonal. And for any w,

$$\left(\sum_{i=1}^{n-k} w_k \tilde{v}_{k+i}\right)^T \left(\sum_{i=1}^{n-k} w_k \tilde{v^*}_{k+i}\right) = \sum_{i=1}^{n-k} w_k^2 (\tilde{v}_{k+i})^T \tilde{v^*}_{k+i} \geq 1 - 8n\gamma^2$$

So we conclude that the distance between these two vectors is at most $3\sqrt{n}\gamma$.

Using this lemma we see that the projected function is close to the projection of f^* in the span of the rest of local maxima (of f^*):

Lemma 6 *Let* g^* *be the projection of* f^* *into the space spanned by the rest of local maxima of* f^**, and* g *be the projection of* f *into the orthogonal subspace of the currently found local maxima, then for any* $\|w\| = 1$ $|g^*(w) - g(w)| \leq \delta/8 + \delta'/20 \leq \delta'/8$.

Proof The proof is straight forward because every vector w correspond to a vector u in the original space S^{n-1} (the u's for different projections are different, but they are close by Lemma 5). Therefore we have

$$|g^*(w) - g(w)| \leq |f^*(u) - f(u)| + |f^*(u) - f^*(u')|$$

for some $\|u - u'\|_2 \leq 3\sqrt{n}\gamma$ (u is the point that corresponds to w under projection for g, u' is the point that corresponds to w under projection for g^*), we know the first one is at most $\delta/8$ and the second one is at most $\delta'/20$ by Lipschitz Condition.

Now we are ready to prove the main theorem.

Proof (Theorem 9) By Theorem 8 the first column is indeed γ close to a local maximum. We then prove by induction that if v_1, v_2, ..., v_k are γ close to different local maxima, then v_{k+1} must be close to a new local maximum.

By Lemma 6 we know g_{k+1} is $(\gamma', \beta', \delta')$ Locally Improvable, and because it is a projection of f its derivatives are also bounded so it is (β', δ') Locally Approximable. By Theorem 8 u' must be γ' close to local maximum for the projected function. Then since the projected space is close to the space spanned by the rest of local maxima, u' is in fact $\gamma' + 3\sqrt{n}\gamma$ close to v^*_{k+1} (here again we are reindexing the local maxima wlog.).

Now we use the Attraction Radius property, since u is currently in U, $f^*(u) \geq T$, and each step we go to a point u' such that $\|u' - u\| \leq \beta$ and $f^*(u') > f^*(u) \geq T$. The local search in Algorithm 2 can never go outside U, therefore it must find the local maximum v^*_{k+1}.

6 Local Search on the Fourth Order Cumulant

Next, we prove that the fourth order cumulant $P^*(u)$ satisfies the properties above. Then the algorithm given in the previous section will find all of the local maxima, which is the missing step in our main goal: learning a noisy linear transformation $Ax + \eta$ with unknown Gaussian noise. We first use a theorem from [16] to show that properties for finding one local maximum are satisfied.

Also, for notational convenience we set $d_i = 2D_A(u_0)^{-2}_{i,i}$ and let $d_{\min}$ and $d_{\max}$ denote the minimum and maximum values (bounds on these and their ratio follow from Claim 6). Using this notation $P^*(u) = 2\sum_{i=1}^n (u^T A_i)^4$ and $P^*(v) = \sum_{i=1}^n d_i v_i^4$ for $v = R^* D_A(u_0)^{-1/2} u$.

Theorem 10 ([16]) *When $\beta < d_{\min}/10 d_{\max} n^2$, the function $P^*(u)$ is*

(a) $(3\sqrt{n}\beta, \beta, P^*(u)\beta^2/100)-$Locally Improvable and

(b) $(\beta, d_{\min}\beta^2/100n)-$Locally Approximable.

Moreover, the local maxima of the function are exactly $\{\pm R_i^*\}$.

Proof The proof appears in [16]. Here for completeness we give the proof using our notation, for completeness. Note that in the proof we will work with 2nd order Taylor's expansion

$$P^*(u) = \text{Proj}_{\perp u}(\nabla P^*(u))^T\xi + \frac{1}{2}\xi^T \text{Proj}_{\perp u}(\mathcal{H}(P^*(u)))\xi - \frac{1}{2}\left(\frac{\partial}{\partial_u}P^*(u)\right)\cdot\|\xi\|_2^2 + O(\beta^3)$$

By the properties of the functions the third order terms are bounded by $O(\beta^3)$ while our improvements will be at least $\Omega(\beta^2)$, so the improvement is still valid for the original function.

First we establish that $P^*(u)$ is Locally Improvable.Observe that this desirada is invariant under orthogonal transformation, so we need only prove the theorem for $P^*(v) = \sum_{i=1}^n d_i v_i^4$. The gradient of the function is $\nabla P^*(v) = 4(d_1v_1^3, d_2v_2^3, ..., d_nv_n^3)$. Then

$$\langle \nabla P^*(v), v\rangle = 4\sum_{i=1}^n d_i v_i^4 = 4P^*(v)$$

Therefore the projected gradient $\phi = \text{Proj}_{\perp v}\nabla P^*(v)$ has coordinate $\phi_i = 4v_i(d_iv_i^2 - P^*(v))$. Furthermore, the Hessian $H = \mathcal{H}(P^*(v))$ is a diagonal matrix whose $(i,i)^{th}$ entry is $12d_iv_i^2$.

Consider the case in which $\|\phi\| \geq P^*(v)\beta/4$. We can obtain an improvement to $P^*(v)\beta^2/100$ because we can take ξ in the direction of ϕ and with $\|\xi\|_2 = \beta/20$. The contribution of the Hessian term is nonnegative and the third term in the Taylor expansion $-2P^*(u)\|\xi\|_2^2$ is small in comparison.

Hence, we can assume $\|\phi\| \leq P^*(v)\beta/4$. Now let us write out the expression of $\|\phi\|^2$

$$\sum_{i=1}^n v_i^2(d_iv_i^2 - P^*(v))^2 \leq \beta^2(P^*(v))^2/16.$$

In particular every term $v_i^2(d_iv_i^2 - P^*(v))^2$ must be at most $\beta^2(P^*(v))^2/16$.. Thus for any i, either $v_i^2 \leq \beta^2$ or $(d_iv_i^2 - P^*(v))^2 \leq (P^*(v))^2/16$.

If there are at least two coordinates k and l such that

$$(d_iv_i^2 - P^*(v))^2 \leq (P^*(v))^2/16$$

then we know for these two coordinates $v_i^2 \in [0.75P^*(v)/d_i, 1.25P^*(v)/d_i]$. We choose the vector ξ so that $\xi_k = \tau v_l$ and $\xi_l = -\tau v_k$. Wlog assume $\xi\cdot\phi \geq 0$ otherwise we use $-\xi$. Take τ so that $\tau^2(v_l^2 + v_k^2) = \beta^2$. Clearly $\|\xi\| = \beta$ and $\xi\cdot v = 0$ so ξ is a valid solution. Also

$$\tau^2 \geq \beta^2/(v_l^2 + v_k^2) \geq \frac{4}{5}\frac{\beta^2}{P^*(u)(1/d_l + 1/d_k)}$$

Now consider the function we are interested in optimizing:

$$\begin{aligned}\phi\cdot\xi + 1/2\xi^T\mathcal{H}\xi - 2P^*(u)\|\xi\|_2 &\geq 1/2\xi^T H\xi - 2P^*(u)\beta^2 \\ &= 6\tau^2v_k^2v_l^2(d_k + d_l) - 2P^*(u)\beta^2 \\ &\geq \frac{27}{8}\tau^2P^*(u)^2\frac{d_k + d_l}{d_kd_l} - 2P^*(u)\beta^2 \geq \frac{7}{10}P^*(u)\beta^2.\end{aligned}$$

In the remaining case, all of the coordinates except for at most one satisfy $v_i^2 \leq \beta^2$. Since we assumed $\beta^2 < \frac{1}{n}$, there must be one of the coordinate v_k that is large, and it is at least $1 - n\beta^2$. Thus the distance of this vector to the local maxima e_k is at most $3\sqrt{n}\beta$.

We then observe that given enough samples, the empirical mean $\widehat{P}'(u)$ is close to $P^*(u)$. For concentration we require every degree four term $z_i z_j z_k z_l$ has variance at most Z.

Claim 11 $Z = O(d_{\min}^2 \lambda_{\min}(A)^8 \|\Sigma\|_2^4 + d_{\min}^2)$.

Proof We will start by bounding $\mathbb{E}[(z_i z_j z_k z_l)^2] \leq \mathbb{E}[(z_i^8 + z_j^8 + z_k^8 + z_l^8)]$. Furthermore $\mathbb{E}[z_i^8] \leq O(\mathbb{E}[(B^{-1}Ax)_i^8 + (B^{-1}\eta)_i^8])$. Note that although here B is the empirical matrix computed in the algorithm, by Lemma 3 it is close to the true quasi-whitening matrix with good probability, here we condition on this event.

Next we bound $\mathbb{E}[(B^{-1}\eta)_i^8]$, which is just the eighth moment of a Gaussian with variance at most $\|B^{-1}\Sigma B^{-T}\|_2 \leq \|B^{-1}\|_2^2 \|\Sigma\|_2 \leq d_{\min}^{1/2}\lambda_{\min}(A)^{-2}\|\Sigma\|_2$. Hence we can bound this term by

$$O(\|B^{-1}\Sigma B^{-T}\|_2^4) = O(d_{\min}^2 \lambda_{\min}(A)^8 \|\Sigma\|_2^4)$$

Finally the remaining term $\mathbb{E}[(B^{-1}Ax)_i^8]$ can be bounded by $O(d_{\min}^2)$ because the variance of this random variable is only larger if we instead replace x by an n-dimensional standard Gaussian.

Lemma 7 *Given $2N$ samples $y_1, y_2, ..., y_N, y'_1, y'_2, ..., y'_N$, suppose columns of $R' = B^{-1}AD_A(u_0)^{1/2}$ are ϵ close to the corresponding columns of R^*, with high probability the function $\widehat{P}'(u)$ is $O(d_{\max}n^{1/2}\epsilon + n^2(N/Z\log n)^{-1/2})$ close to the true function $P^*(u)$.*

Proof $\widehat{P}'(u)$ is the empirical mean of

$$F(u, y, y') = -(u^T B^{-1} y)^4 + 3(u^T B^{-1} y)^2 (u^T B^{-1} y')^2$$

In Section 3 we proved that $P'(u) = \mathbb{E}_{y,y'} F(u, y, y') = \sum_{i=1}^n 2D_{i,i}^{-1/2}(u^T R'_i)^4 = \sum_{i=1}^n d_i (u^T R'_i)^4$. First, we demonstrate that $P'(u)$ is close to $P^*(u)$, and then using concentration bounds we show that $\widehat{P}'(u)$ is close to $P'(u)$ (with high probability) over all u.

The first part is a simple application of Cauchy-Schwartz:

$$|P'(u) - P^*(u)| = \sum_{i=1}^n d_i \left[(u^T R'_i) - (u^T R_i^*)\right] \cdot \left[(u^T R'_i + u^T R_i^*)((u^T R'_i)^2 + (u^T R_i^*)^2)\right]$$

$$\leq d_{\max} \sqrt{\sum_{i=1}^n (u^T(R'_i - R_i^*))^2} \cdot (3 \left\|u^T R' + u^T R^*\right\|_2) \leq 6 d_{\max} n^{1/2} \epsilon.$$

The first inequality uses the fact that $((u^T R'_i)^2 + (u^T R^*_i)^2) \leq 3$, the second inequality uses the fact that when ϵ is small enough, $\left\|u^T R'\right\|_2 \leq 2$.

Next we prove that the empirical mean $\widehat{P}'(u)$ is close to $P'(u)$. The key point here is we need to prove this for all points u since a priori we have no control over which directions local search will choose to explore. We accomplish this by considering $\widehat{P}'(u)$ as a degree-4 polynomial over u and prove that the coefficient of each monomial in $\widehat{P}'(u)$ is close to the corresponding coefficient in $P'(u)$. This is easy: the expectation of each coefficient of $F(u, y, y')$ is equal to the correct coefficient, and the variance is bounded by $O(Z)$. The coefficients are also sub-Gaussian so by Bernstein's inequality the probability that any coefficient of $\widehat{P}'(u)$ deviates by more than ϵ' (from its expectation) is at most $e^{-\Omega(\epsilon'^2 N/Z)}$. Hence when $N \geq O(Z \log n/\epsilon'^2)$ with high probability all the coefficients of $\widehat{P}'(u)$ and $P'(u)$ are ϵ' close. When we write

$$P'(u) = \sum_{i_1,i_2,i_3,i_4=1}^{n} P'_{i_1,i_2,i_3,i_4} u_{i_1} u_{i_2} u_{i_3} u_{i_4}$$

$$\widehat{P}'(u) = \sum_{i_1,i_2,i_3,i_4=1}^{n} \widehat{P}'_{i_1,i_2,i_3,i_4} u_{i_1} u_{i_2} u_{i_3} u_{i_4}$$

And for any u:

$$\begin{aligned} |P'(u) - \widehat{P}'(u)| &\leq \sum_{i_1,i_2,i_3,i_4=1}^{n} |(P'_{i_1,i_2,i_3,i_4} - \widehat{P}'_{i_1,i_2,i_3,i_4}) u_{i_1} u_{i_2} u_{i_3} u_{i_4}| \\ &\leq \epsilon' (\sum_{i=1}^{n} |u_i|)^4 \leq \epsilon' n^2. \end{aligned}$$

Therefore $\widehat{P}'(u)$ and $P^*(u)$ are $O(d_{\max} n^{1/2} \epsilon + n^2 (N/Z \log n)^{-1/2})$ close.

This proof can also be used to show that the derivatives of the function $\widehat{P}'(u)$ is concentrated to the derivatives of the true function $P^*(u)$ because the derivatives are only related to coefficients. Since we know $P^*(u)$ is $(\beta, d_{\min}\beta^2/100n)-$Locally Approximable (Theorem 10), when we take ϵ to be small enough and N to be large enough (both polynomial in the parameters), we have $\widehat{P}'(u)$ is also $(\beta, d_{\min}\beta^2/50n)-$Locally Approximable.

The other properties required by Theorem 9 are also satisfied:

Lemma 8 *For any $\|u - u'\|_2 \leq r$, $|P^*(u) - P^*(u')| \leq 5 d_{\max} n^{1/2} r$. All local maxima of P^* has attraction radius $Rad \geq d_{\min}/100 d_{\max}$.*

Proof The Lipschitz condition follows from the same Cauchy-Schwartz as appeared above. When two points u and u' are of distance r, $|P^*(u) - P^*(u')| \leq 5 d_{\max} n^{1/2} r$. Finally for the Attraction Radius, we know when $3\sqrt{n}\gamma + \gamma' \leq d_{\min}/100 d_{\max}$, we can just take the set U to be $u^T R^*_i \geq 1 - d_{\min}/50 d_{\max}$. For all u such that $u^T R^*_i \in [1 - d_{\min}/25 d_{\max}, 1 - d_{\min}/50 d_{\max}]$ (which contains the β neighborhood of U), we know the value of $P^*(u) \leq T$.

Algorithm 4. RECOVER, **Input:** B, $\widehat{P}'(u)$, $\widehat{R}$, ϵ **Output:** $\widehat{A}$, $\widehat{\Sigma}$

1. Let $\widehat{D}_A(u)$ be a diagonal matrix whose i^{th} entry is $\frac{1}{2}\left(\widehat{P}'(\widehat{R}_i)\right)^{-1/2}$.
2. Let $\widehat{A} = B\widehat{R}\widehat{D}_A(u)^{-1/2}$.
3. Estimate $C = \mathbb{E}[yy^T]$ by taking $O((\|A\|_2 + \|\Sigma\|_2)^4 n^2 \epsilon^{-2})$ samples and let $\widehat{C} = \frac{1}{N}\sum_{i=1}^{N} y_i y_i^T$.
4. Let $\widehat{\Sigma} = \widehat{C} - \widehat{A}\widehat{A}^T$
5. Return $\widehat{A}, \widehat{\Sigma}$

Applying Theorem 9 we obtain the following Lemma (the parameters are chosen so that all properties required are satisfied):

Lemma 9 *Let $\beta' = \Theta((d_{\min}/d_{\max})^2)$, $\beta = \min\{\gamma n^{-1/2}, \Omega((d_{\min}/d_{\max})^4 n^{-3.5})\}$, then the procedure* RECOVER$(f, \beta,\ d_{\min}\beta^2/100n\ , \beta', d_{\min}\beta'^2/100n)$ *finds vectors $v_1, v_2, ..., v_n$, so that there is a permutation matrix Π and $s_i \in \{\pm 1\}$ and for all i: $\|v_i - (R\Pi Diag(s_i))_i^*\|_2 \leq \gamma$.*

After obtaining $\widehat{R} = [v_1, v_2, ..., v_n]$ we can use Algorithm 4 to find A and Σ:

Theorem 12 *Given a matrix $\widehat{R}$ such that there is permutation matrix Π and $s_i \in \{\pm 1\}$ with $\|\widehat{R}_i - s_i(R^*\Pi)_i\|_2 \leq \gamma$ for all i, Algorithm 4 returns matrix $\widehat{A}$ such that*

$$\|\widehat{A} - A\Pi Diag(s_i)\|_F \leq O(\gamma \|A\|_2^2 n^{3/2}/\lambda_{\min}(A))$$

Moreover if $\gamma \leq O(\epsilon/\|A\|_2^2 n^{3/2}\lambda_{\min}(A)) \times \min\{1/\|A\|_2, 1\}$, we also have $\|\widehat{\Sigma} - \Sigma\|_F \leq \epsilon$.

Recall that the diagonal matrix $D_A(u)$ is unknown (since it depends on A), but if we are given R^* (or an approximation) and since $P^*(u) = \sum_{i=1}^n d_i (u^T R_i^*)^4$, we can recover the matrix $D_A(u)$ approximately from computing $P^*(R_i^*)$. Then given $D_A(u)$, we can recover A and Σ and this completes the analysis of our algorithm.

Proof By Lemma 4 we know the columns of $\widehat{R}$ is close the the columns of R^* (the parameters will be set so that the error is much smaller than γ), thus $\|\widehat{R}_i - s_i(R'\Pi)_i\|_2 \leq \gamma$. Applying Lemma 7 we obtain: $|\widehat{P}'(\widehat{R}_i) - P^*(\widehat{R}_i)| \ll \gamma$. Furthermore, when $\|\widehat{R}_i - s_i R^*_{\Pi^{-1}(i)}\|_2 \leq \gamma$ we know that $P^*(\widehat{R}_i)/d_{\Pi^{-1}(i)} \in [1-3\gamma, 1+3\gamma]$ (here we are abusing notation and use the permutation matrix as a permutation). Hence $\widehat{D}_A(u)_{i,i}/\left(D_A(u)\right)_{\Pi^{-1}(i),\Pi^{-1}(i)} \in [1-3\gamma, 1+3\gamma]$. We have:

$$\widehat{A}_i = B\widehat{R}_i \widehat{D}_A(u)_{i,i}^{-1/2} \text{ and } \left(A\Pi \text{Diag}(s_i)\right)_i = BR'_{\Pi^{-1}(i)} \left(D_A(u)\right)^{-1/2}_{\Pi^{-1}(i),\Pi^{-1}(i)}$$

and their difference is at most $O(\gamma \|B\|_2 \left(D_A(u)\right)^{-1/2}_{\Pi^{-1}(i),\Pi^{-1}(i)})$. Hence we can bound the total error by $O(\gamma \|B\|_2 \left\|D_A(u)^{-1/2}\right\|_F)$. We also know $\|B\|_2 \leq \|A\|_2 \|D_A(u)^{1/2}\|_2$ because $BB^T \approx AD_A(u)A^T$, so this can be bounded by

$O(\gamma \|A\|_2 \|D_A(u)\|_2^{1/2} \|D_A(u)^{-1/2}\|_F)$. Applying Claim 6, we conclude that (with high probability) the ratio of the largest to smallest diagonal entry of $D_A(u)$ is at most $9n^3 \log n \|A\|_2^2 / \lambda_{\min}(A)^2$ (because $\max \|A_i\|^2 \leq \|A\|_2^2$ and $\min \|A_i\|^2 \geq \lambda_{\min}(A)^2$). So we can bound the error by

$$O(\gamma \|A\|_2^2 n^{5/2} \log n / \lambda_{\min}(A))$$

Consider the error for Σ: Using concentration bounds similar but much simpler than those used in Lemma 7, we obtain that $\|\widehat{C} - C\|_F \leq \epsilon/2$. On the other hand, $\|\widehat{A}\widehat{A}^T - AA^T\|_F = \|\widehat{A}\widehat{A}^T - A\Pi \mathrm{Diag}(s_i)(A\Pi \mathrm{Diag}(s_i))^T\|_F \leq 2\|A\|_2 \|A\Pi \mathrm{Diag}(s_i) - \widehat{A}\|_F + \|A\Pi \mathrm{Diag}(s_i) - \widehat{A}\|_F^2 \leq \epsilon/2$ (when γ is a suitably small polynomial in the parameters). Therefore $\|\widehat{\Sigma} - \Sigma\|_F \leq \|\widehat{C} - C\|_F + \|\widehat{A}\widehat{A}^T - AA^T\|_F \leq \epsilon$. This completes the proof of the theorem.

Conclusions

Independent Component Analysis is a vast field with many successful techniques. Most rely on heuristic nonlinear optimization. An exciting question is: Can we give a rigorous analysis of those techniques as well, just as we did for local search on cumulants? A rigorous analysis of deep learning — say, an algorithm that provably learns the parameters of a Restricted Boltzmann Machine — is another problem that is wide open, and a plausible special case involves subtle variations on the problem we considered here.

References

1. A. Anandkumar, D. Foster, D. Hsu, S. Kakade, Y. Liu. Two SVDs suffice: spectral decompositions for probabilistic topic modeling and latent dirichlet allocation. *Arxiv:*abs/1203.0697, 2012.
2. J. Anderson, M. Belkin, N. Goyal, L Rademacher and J. Voss. The more the merrier: the blessing of dimensionality for learning large gaussian mixtures. *arxiv:1311.2891*, 2013.
3. S. Arora and R. Kannan. Learning mixtures of separated nonspherical gaussians. *Annals of Applied Probability*, pp. 69-92, 2005.
4. M. Belkin, L. Rademacher and J. Voss. Blind signal separation in the presence of gaussian noise. In *COLT* 2013.
5. M. Belkin and K. Sinha. Polynomial learning of distribution families. *FOCS* pp. 103–112, 2010.
6. Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, pp. 1–127, 2009.
7. A. Bhaskara, M. Charikar and A. Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability. *arxiv:1304.8087*, 2013.
8. A. Bhaskara, M. Charikar, A. Moitra and A. Vijayaraghavan. Smoothed analysis of tensor decompositions. *arxiv:1311.3651*, 2013.
9. P. Comon. Independent component analysis: a new concept? *Signal Processing*, pp. 287–314, 1994.
10. S. Cruces, L. Castedo, A. Cichocki, Robust blind source separation algorithms using cumulants, *Neurocomputing*, Volume 49, Issues 14, pp 87-118, 2002.
11. S. Dasgupta. Learning mixtures of Gaussians. *FOCS* pp. 634–644, 1999.

12. L. De Lathauwer, J Castaing and J. Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Trans. on Signal Processing*, 55(6):2965–2973, 2007.
13. L., De Lathauwer; B., De Moor; J. Vandewalle. Independent component analysis based on higher-order statistics only *P*roceedings of 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing, 1996.
14. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM Algorithm. *Journal of the Royal Statistical Society Series B*, pp. 1–38, 1977.
15. Y. Freund , D. Haussler. Unsupervised learning of distributions on binary vectors using two layer networks University of California at Santa Cruz, Santa Cruz, CA, 1994
16. A. Frieze, M. Jerrum, R. Kannan. Learning linear transformations. *FOCS*, pp. 359–368, 1996.
17. N. Goyal, S. Vempala and Y. Xiao. Fourier PCA. *arxiv:1306.5825*, 2013.
18. G. E. Hinton. A practical guide to training restricted boltzmann machines. UTML TR 2010-003, Department of Computer Science, University of Toronto, August 2010
19. G. Hinton, R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science* pp. 504–507, 2006.
20. D. Hsu, S. Kakade. Learning mixtures of spherical Gaussians: moment methods and spectral decompositions. *Arxiv:*abs/1206.5766, 2012.
21. P. J. Huber. Projection pursuit. *Annals of Statistics* pp. 435–475, 1985.
22. A. Hyvarinen, J. Karhunen, E. Oja. *Independent Component Analysis.* Wiley: New York, 2001.
23. A. Hyvarinen, E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, pp. 1483–1492, 1997.
24. A. Hyvarinen, E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, pp. 411–430, 2000.
25. A. T. Kalai, A. Moitra, and G. Valiant. Efficiently learning mixtures of two Gaussians. *STOC* pp. 553-562, 2010.
26. M. Kendall, A. Stuart. *The Advanced Theory of Statistics* Charles Griffin and Company, 1958.
27. A. Moitra and G. Valiant. Setting the polynomial learnability of mixtures of Gaussians. *FOCS* pp. 93–102, 2010.
28. S. Vempala, Y. Xiao. Structure from local optima: learning subspace juntas via higher order PCA. *Arxiv:*abs/1108.3329, 2011.
29. A. Yeredor. Blind source separation via the second characteristic function. *Signal Processing*, pp. 897–902, 2000.