Homotopy Analysis for Tensor PCA

Anima Anandkumar* Yuan Deng† Rong Ge‡ Hossein Mobahi§

November 3, 2016

Abstract

Developing efficient and guaranteed nonconvex algorithms has been an important challenge in modern machine learning. Algorithms with good empirical performance such as stochastic gradient descent often lack theoretical guarantees. In this paper, we analyze the class of homotopy or continuation methods for global optimization of nonconvex functions. These methods start from an objective function that is efficient to optimize (e.g. convex), and progressively modify it to obtain the required objective, and the solutions are passed along the homotopy path. For the challenging problem of tensor PCA, we prove global convergence of the homotopy method in the "high noise" regime. The signal-to-noise requirement for our algorithm is tight in the sense that it matches the recovery guarantee for the *best* degree-4 sum-of-squares algorithm. In addition, we prove a phase transition along the homotopy path for tensor PCA. This allows to simplify the homotopy method to a local search algorithm, viz., tensor power iterations, with a specific initialization and a noise injection procedure, while retaining the theoretical guarantees.

Keywords: Tensor PCA, homotopy, continuation, Gaussian smoothing, nonconvex optimization, global optimization.

1 Introduction

Non-convex optimization is a critical component in modern machine learning. Simple local-search algorithms such as stochastic gradient descent have enjoyed great empirical success in areas such as deep learning. However, theoretical guarantees for nonconvex optimization have been mostly negative, and the problems are computationally hard in the worst case. Recent research efforts have attempted to bridge this gap between theory and practice.

It has been recently proven that for many well known nonconvex problems, all local optima are also global optima, under mild conditions. Consequently, local-search methods, which are designed to find a local optimum, automatically achieve global optimality. Examples of such problems include matrix completion [10], orthogonal tensor decomposition [2, 9], phase retrieval [22], complete dictionary learning [21], and so on. However, such a class of nonconvex problems is limited, and there are many practical problems with poor local optima, where local search methods can fail.

*University of California, Irvine. Email:a.anandkumar@uci.edu

†Duke University. Email:ericdy@cs.duke.edu

‡Duke University. Email:rongge@cs.duke.edu

§Google Research. Email:hmobahi@csail.mit.edu

A milder requirement for the success of local search methods is the ability to initialize in the basin of attraction of the global optimum using another polynomial-time algorithm. Thus, this approach does not require all the local optima to be of good quality. Efficient initialization strategies have recently been developed for many nonconvex problems such as overcomplete dictionary learning [4, 1] and tensor decomposition [3], robust PCA [18], mixed linear regression [23] and so on.

Although the list of such tractable nonconvex problems is growing, so far, the initialization algorithms are problem-specific and as such, cannot be directly extended to new problems. An interesting question is whether there exist universal principles that can be used in designing such efficient initialization schemes for local search methods. In this paper, we demonstrate how the class of homotopy or continuation methods can provide such a framework for efficient initialization of local search methods.

The homotopy method starts from an objective function that is efficient to optimize (e.g. convex function), and progressively transforms it to the required objective [16]. Throughout this progression, the solution of each intermediate objective is used to initialize a local search on the next one. A popular approach for constructing this progression is to smooth the objective function. Precisely, the objective function is convolved with the Gaussian kernel and the amount of smoothing is varied to obtain the set of transformations. Intuitively, smoothing "erases wiggles" on the objective surface (which can lead to poor local optima), thereby resulting in a function that is easier to optimize. Global optimality guarantees for the homotopy method have been recently established [17, 11]. However, the assumptions in these results are either too restrictive [17] or extremely difficult to check [11]. In addition, homotopy algorithms are generally slow since local-search is repeated within each instantiation of the smoothed objective.

In this paper, we address all the above issues for the nonconvex tensor PCA problem. We analyze the homotopy method and guarantee convergence to global optimum under a set of transparent conditions. Additionally, we demonstrate how the homotopy method can be simplified without sacrificing the theoretical guarantees. We show that by taking advantage of the phase transitions in the homotopy path, we can avoid the intermediate transformations of the objective function. In fact, we can start from the extreme case of "easy" (convex) function, and use its solution to initialize local search on the original objective. Thus, we show that the homotopy method can serve as a principle for obtaining a smart initialization which is then employed in efficient local search methods. Although we limit ourselves to the problem of tensor PCA in this paper, we expect the developed techniques to be applicable for other nonconvex problems.

Tensor PCA problem is an extension of the matrix PCA. The statistical model for tensor PCA was first introduced by [20]. This is a single spike model where the input tensor $\boldsymbol{T} \in \mathbb{R}^{n\times n\times n}$ is a combination of an unknown rank-1 tensor and a Gaussian noise tensor $\boldsymbol{A}$ with $\boldsymbol{A}_{ijk} \sim \mathcal{N}(0,1)$ for $i, j, k \in [n]$.

$$\boldsymbol{T} = \lambda \boldsymbol{v} \otimes \boldsymbol{v} \otimes \boldsymbol{v} + \boldsymbol{A}, \tag{1}$$

where $\boldsymbol{v} \in \mathbb{R}^n$ is the signal that we would like to recover.

Tensor PCA belongs to the class of "needle in a haystack" or high dimensional denoising problems, where the goal is to separate the unknown signal from a large amount of random noise. Recovery in the high noise regime has intimate connections to computational hardness, and has been extensively studied in a number of settings. For instance, in the spiked random matrix model, the input is an additive combination of an unknown rank-1 matrix and a random noise matrix. The requirement on the signal-to-noise ratio for simple algorithms, such as principal component

Method	Bound on τ	Time	Space
Power method + initialization + noise injection (ours)	$\tilde{\Omega}(n^{3/4})$	$\tilde{O}(n^3)$	$\tilde{O}(n)$
Power method, random initialization	$\tilde{\Omega}(n)$	$\tilde{O}(n^3)$	$\tilde{O}(n)$
Sum-of-Squares	$\tilde{\Omega}(n^{3/4})$	$> \Omega(n^6)$	$> \Omega(n^6)$
Recovery and Certify	$\tilde{\Omega}(n^{3/4})$	$\tilde{O}(n^5)$	$O(n^4)$
Eigendecomposition of flattened matrix	$\tilde{\Omega}(n^{3/4})$	$\tilde{O}(n^3)$	$\tilde{O}(n^2)$
Information-theoretic	$\tilde{\Omega}(\sqrt{n})$	Exp	$O(n)$

Table 1: Table of comparison of various methods for tensor PCA. Here space does not include the tensor itself. The power method with random initialization was analyzed in [20]. sum-of-squares, Recover and Certify, and flattened tensor were analyzed in [13].

analysis (PCA), to recover the unknown signal has been studied under various noise models [19, 6] and sparsity assumptions on the signal vector [5].

Previous algorithms for tensor PCA belong to two classes: local search methods such as tensor power iterations [20], and global methods such as sum of squares [13]. Currently, the best signal-to-noise guarantee is achieved by the sum-of-squares algorithm and the flattening algorithm, which are more expensive compared to power iterations (see Table 1). In this paper, we analyze the Gaussian homotopy method for tensor PCA, and prove that it matches the best known signal-to-noise performance. [13] also showed a lowerbound that no degree-4 (or lower) sum-of-squares algorithm can achieve better signal-to-noise ratio, implying that our analysis is likely to be tight.

We also prove a phase transition along the homotopy path for the tensor PCA problem. This implies that we can skip intermediate steps in the homotopy path, and obtain a simpler algorithm with the same guarantees as the homotopy method. We are thus able to modify the homotopy algorithm to obtain a simple variant of the tensor power iterations with a specific initialization and simple noise injection mechanism while retaining the signal-to-noise performance. Thus, we obtain simple and efficient algorithm with tight performance for the tensor PCA problem.

Our Results We analyze a simple variant of the popular tensor power method, which is a local search method for finding the best rank-1 approximation of the input tensor. We modify it by using a specific initialization and inject appropriate random noise in each iteration. This runs almost in linear time; see Table 1 for more details.

Theorem 1.1 (informal)**.** *There is an almost linear time algorithm for tensor PCA that finds the signal $\boldsymbol{v}$ as long as the signal strength $\tau = \tilde{\Omega}(n^{3/4})$.*

Our algorithm achieves the best possible trade-offs among all known algorithms (see Table 1). The algorithm and its analysis utilizes the framework of homotopy. We establish a phase transition along the homotopy path.

Theorem 1.2 (informal)**.** *There is a threshold θ such that if the radius of smoothing is significantly larger than θ, the smoothed function will have a unique local and global maximum. If the radius of smoothing is smaller, then the smoothed function can have multiple local maxima, but one of them is close to the signal vector $\boldsymbol{v}$.*

The above result allows us to skip the intermediate steps in the homotopy path. We only need two end points of the homotopy path: the original objective function with no smoothing and with an infinite amount of smoothing. The optimal solution for the latter can be obtained through any local search method; in fact, in our case, it has a closed form. This serves as initialization for the original objective function.

To prove the above theorem we require an Independence Assumption, which roughly says the noise tensor $\boldsymbol{A}$ does not adversarially correlate with the signal $\boldsymbol{v}$ for all points on the homotopy path. We give a noise injection procedure under which this assumption holds for our algorithm. This is standard in analyzing nonconvex optimization algorithms. As an example, previous works on alternating minimization for matrix completion [14] relied on the availability of different subsamples in different iterations to obtain the theoretical guarantees. Our noise injection procedure is very similar, however this is the first application of this idea for the case of Gaussian noise.

The comparison of all the current algorithms for tensor PCA is given in Table 1. Note that the space in the table does not include the space for storing the tensor, this is because the more practical algorithms only access the tensor for a very small number of passes, which allows the algorithms to be implemented online and do not need to keep the whole tensor in the memory. We see that our algorithm has the best performance across all the measures. In our synthetic experiments (see Section 5, we find that our method significantly outperforms the other methods: it converges to a better solution faster and with a lower variance.

2 Preliminaries

In this section, we formally define the tensor PCA problem and its objective functions. Then we show how to compute the smoothed versions of these objective functions.

2.1 Tensors and Polynomials

Tensors are higher dimensional generalizations of matrices. In this paper we focus on 3rd order tensor, which corresponds to a 3 dimensional array. Given a vector $\boldsymbol{v} \in \mathbb{R}^n$, similar as rank one matrices $\boldsymbol{v}\boldsymbol{v}^\top$, we consider rank 1 tensors $\boldsymbol{v}^{\otimes 3}$ to be a $n \times n \times n$ array whose i, j, k-th entry is equal to $\boldsymbol{v}_i \boldsymbol{v}_j \boldsymbol{v}_k$.

For a matrix $\boldsymbol{M}$, we often consider the quadratic form it defines: $\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}$. Similarly, for a tensor $\boldsymbol{T} \in \mathbb{R}^{n\times n\times n}$, we define a degree 3 polynomial $\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) = \sum_{i,j,k=1}^n \boldsymbol{T}_{i,j,k} \boldsymbol{x}_i \boldsymbol{x}_j \boldsymbol{x}_k$.

This polynomial is just a special form of a trilinear form defined by the tensor. Given three vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$, the trilinear form $\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \sum_{i,j,k=1}^n \boldsymbol{T}_{i,j,k} \boldsymbol{x}_i \boldsymbol{y}_j \boldsymbol{z}_k$. Using this trilinear form, we can also consider the tensor as an operator that maps vectors to matrices, or two vectors into a single vector. In particular, $\boldsymbol{T}(\boldsymbol{x}, :, :)$ is a matrix whose i, j-th entry is equal to $\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{e}_i, \boldsymbol{e}_j)$ where $\boldsymbol{e}_i$ is the i-th basis vector. Similarly, $\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{y}, :)$ is a vector whose i-th coordinate is equal to $\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{e}_i)$.

Since the tensor $\boldsymbol{T}$ we consider is not symmetric ($\boldsymbol{T}_{ijk}$ is not necessarily equal to $\boldsymbol{T}_{jik}$ or other permutations), we also define the symmetric operator

$$\delta(\boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x}, :) + \boldsymbol{T}(\boldsymbol{x}, :, \boldsymbol{x}) + \boldsymbol{T}(:, \boldsymbol{x}, \boldsymbol{x}).$$

2.2 Objective Functions for Tensor PCA

We first define the tensor PCA problem formally.

Definition 1 (Tensor PCA)**.** Given input tensor $\boldsymbol{T} = \tau \cdot \boldsymbol{v}^{\otimes 3} + \boldsymbol{A}$, where $\boldsymbol{v} \in \mathbb{R}^n$ is an arbitrary unit vector, $\tau \geq 0$ is the signal-to-noise ratio, and $\boldsymbol{A}$ is a random noise tensor with iid standard Gaussian entries, recover the signal $\boldsymbol{v}$ approximately (find a vector $\|\boldsymbol{x}\| = 1$ such that $\langle \boldsymbol{x}, \boldsymbol{v} \rangle \geq 1/2$).

Similar to the Matrix PCA where we maximize the quadratic form, for tensor PCA we can focus on optimizing the degree 3 polynomial $f(\boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x})$ over the unit sphere.

$$\begin{aligned} \max \quad & f(\boldsymbol{x}) = \tau \langle \boldsymbol{v}, \boldsymbol{x} \rangle^3 + \boldsymbol{A}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) \\ & \|\boldsymbol{x}\| = 1 \end{aligned} \tag{2}$$

The optimal value of this program is known as the spectral norm of the tensor. It is often solved in practice by tensor power method. [20] noticed that:

Theorem 2.1. *When $\tau \geq C\sqrt{n}$ for large constant C, the global optimum of (2) is close to the signal $\boldsymbol{v}$.*

However, solving this optimization problem is NP-hard in the worst-case[12]. Previously, the best known algorithm uses sum-of-squares hierarchy and works when $\tau \geq Cn^{3/4}$. There is a huge gap between what's achievable information theoretically ($O(\sqrt{n})$) and what can be achieved algorithmically ($O(n^{3/4})$).

2.3 Gaussian Smoothing for the Objective Function

Guaranteed homotopy methods rely on smoothing the objective function by the Gaussian kernel [16, 17]. More precisely, smoothing the objective (2) requires convolving it with the Gaussian kernel. Let $g : \mathcal{X} \times \mathbb{R}^+ \to \mathbb{R}$ be a mapping such that

$$g(\boldsymbol{x}, t) = [f \star k_t](\boldsymbol{x})$$

Here, k_t is the Gaussian density function for $\mathcal{N}(\boldsymbol{0}, t^2 \boldsymbol{I}_n)$, satisfying

$$k_t(\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi}t)^n} \cdot e^{\frac{-\|\boldsymbol{x}\|_2^2}{2t^2}}.$$

Since our objective function $f(\boldsymbol{x})$ is a polynomial, we can compute the closed form of $g(\boldsymbol{x}, t)$.

Lemma 1 (Smoothed Tensor PCA Objective)**.** *The smoothed objective has the form*

$$g(\boldsymbol{x}, t) = \tau \langle \boldsymbol{v}, \boldsymbol{x} \rangle^3 + t^2 \langle 3\tau \boldsymbol{v} + \boldsymbol{u}, \boldsymbol{x} \rangle + \boldsymbol{A}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}),$$

where the vector $\boldsymbol{u}$ is defined by $\boldsymbol{u}_j = \sum_{i=1}^n (\boldsymbol{A}_{iij} + \boldsymbol{A}_{iji} + \boldsymbol{A}_{jii})$. Moreover, it is easy to compute vector $\boldsymbol{z} = 3\tau \boldsymbol{v} + \boldsymbol{u}$ given just the tensor $\boldsymbol{T}$, as $\forall j \quad \boldsymbol{z}_j = \sum_{i=1}^n (\boldsymbol{T}_{iij} + \boldsymbol{T}_{iji} + \boldsymbol{T}_{jii})$.

The proof of this Lemma is based on interpreting the convolution as an expectation $\mathbb{E}_{y \sim N(\mathbf{0}, \mathbf{I}_n)}[f(x + y)]$. We defer the detailed calculation to Appendix **??**

3 Tensor PCA by Homotopy Initialization

In this section we give a simple smart initialization algorithm for tensor PCA. Our algorithm only uses two points in homotopy path − the infinite smoothing $t \to \infty$ and the no smoothing $t \to 0$. This is inspired by our full analysis of the homotopy path (see Section 4), where we show there is a *phase transition* in the homotopy path. When the smoothing parameter is larger than a threshold, the function behaves like the infinite smoothing case; when the smoothing parameter is smaller than the threshold, the function behaves like the no smoothing case.

Recall that the smoothed function $g(\boldsymbol{x}, t)$ is:

$$g(\boldsymbol{x}, t) = \tau \langle \boldsymbol{v}, \boldsymbol{x} \rangle^3 + t^2 \langle 3\tau\boldsymbol{v} + \boldsymbol{u}, \boldsymbol{x} \rangle + \boldsymbol{A}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) \tag{3}$$

with $\boldsymbol{u}$ as a vector such that $\boldsymbol{u}_j = \sum_i \boldsymbol{A}_{iij} + \boldsymbol{A}_{iji} + \boldsymbol{A}_{jii}$. When $t \to \infty$, the solution of the smoothed problem has the special form $\boldsymbol{x}^\dagger = \frac{3\tau\boldsymbol{v}+\boldsymbol{u}}{\|3\tau\boldsymbol{v}+\boldsymbol{u}\|}$. That is because the term $t^2\langle 3\tau\boldsymbol{v} + \boldsymbol{u}, \boldsymbol{x}\rangle$ dominates g and thus its maximizer under $\|\boldsymbol{x}\|_2 = 1$ yields $\boldsymbol{x}^\dagger$.

Note that by Lemma 1, although we only know $\boldsymbol{T}$, we can actually compute this vector by normalizing $\boldsymbol{z}_j = \sum_{i=1}^n \boldsymbol{T}_{iij} + \boldsymbol{T}_{iji} + \boldsymbol{T}_{jii}$. We use this point as an initialization, and then run power method on the original function. The resulting algorithm is described in Algorithm 1.

In order to analyze the algorithm, we assume the following *independence* assumption:

Assumption 1. [Independence Assumption] For the sequence computed by Algorithm 1, $\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^m$, assume that for all $0 \le p \le m$, with high probability (1) $\|\delta(\boldsymbol{x}^p)\|_2 = \Theta(\sqrt{nm})\|\boldsymbol{x}^p\|_2^2$, (2) $\langle \delta(\boldsymbol{x}^p), \boldsymbol{v}\rangle = O(\sqrt{\log m})\|\boldsymbol{x}^p\|_2^2$, (3) $\|\boldsymbol{u}\|_2 = \Theta(n\sqrt{m})$, and (4) $\langle \boldsymbol{u}, \boldsymbol{v}\rangle = O(\sqrt{nm \log m})$.

Intuitively, all of these conditions are easily satisfied if in every step of the algorithm, the noise tensor $\boldsymbol{A}$ is *resampled* to be a fresh random matrix. This is of course not really true, because the points $\boldsymbol{x}^i$'s are dependent on $\boldsymbol{A}$. However, we are able to modify the algorithm by a *noise injection* procedure, that adds more noise to the tensor $\boldsymbol{T}$, and make the noise tensor "look" as if they were independent. We will first show the correctness of the algorithm assuming independence here, and in Section 3.1 we discuss the noise injection procedure.

Theorem 3.1. *When $\tau = \tilde{\Omega}(n^{3/4})$, Algorithm 1 finds a vector $\boldsymbol{x}^m$ that is close to $\boldsymbol{v}$ in $O(\log n/\log\log n)$ iterations.*

Algorithm 1: Tensor PCA by Homotopy Initialization

```
Input: Tensor $\boldsymbol{T} = \tau \cdot \boldsymbol{v}^{\otimes 3} + \boldsymbol{A}$;
Output: Approximation of $\boldsymbol{v}$;
$m = O(\log n/\log\log n)$;
$\forall\ j, \boldsymbol{x}_j^0 = \sum_i \boldsymbol{T}_{iij} + \boldsymbol{T}_{iji} + \boldsymbol{T}_{jii}$;
$\boldsymbol{x}^0 = \boldsymbol{x}^0/\|\boldsymbol{x}^0\|$;                    // Now $\boldsymbol{x}^0 = \boldsymbol{x}^\dagger$
for $k = 0$ to $m$ do
    $\boldsymbol{x}^{k+1} = \boldsymbol{T}(\boldsymbol{x}^k, \boldsymbol{x}^k, :) + \boldsymbol{T}(\boldsymbol{x}^k, :, \boldsymbol{x}^k) + \boldsymbol{T}(:, \boldsymbol{x}^k, \boldsymbol{x}^k)$;
    $\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k+1}/\|\boldsymbol{x}^{k+1}\|$;
return $\boldsymbol{x}^m$;
```

The main idea is to show the correlation of $\boldsymbol{x}^k$ and $\boldsymbol{v}$ increases in every step. In order to do this, first notice that the initial point $\boldsymbol{x}^\dagger$ itself is equal to a normalization of $3\tau\boldsymbol{v} + \boldsymbol{u}$, where the norm of $\boldsymbol{u}$ and its correlation with $\boldsymbol{v}$ are all bounded by the Independence Assumption. It is easy to check that $\langle \boldsymbol{x}^0, \boldsymbol{v}\rangle \gg n^{-1/4}$, which is already non-trivial because a random vector would only have correlation around $n^{1/2}$. For the later iterations, let $\hat{\boldsymbol{x}}^k$ be the vector $\boldsymbol{x}^k$ before normalization, notice that $\hat{\boldsymbol{x}}^{k+1} = 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^k\rangle^2 \boldsymbol{v} + \delta(\boldsymbol{x}^k)$. Notice that the first term is in the direction $\boldsymbol{v}$, and the Independence Assumption bounds the norm and correlation with $\boldsymbol{v}$ for the second term. We can show that the correlation with $\boldsymbol{v}$ increases in every iteration, because the initial point already has a large inner product with $\boldsymbol{v}$. The detailed proof is deferred to the supplementary material.

3.1 Noise Injection Procedure

Algorithm 2: Tensor PCA with Homotopy Initialization and Noise Injection

```
Input: Tensor $\boldsymbol{T} = \tau \cdot \boldsymbol{v}^{\otimes 3} + \boldsymbol{A}$;
Output: Approximation of $\boldsymbol{v}$;
$m = O(\log n / \log\log n)$;
Sample $\boldsymbol{A}^0, \boldsymbol{A}^1, ..., \boldsymbol{A}^{m-1} \in \mathbb{R}^{n\times n\times n}$ whose entries are $\mathcal{N}(0, m)$.
Let $\overline{\boldsymbol{A}} = \frac{1}{m}\sum_{i=0}^{m-1} \boldsymbol{A}^i$.
Let $\boldsymbol{T}^i = \boldsymbol{T} - \overline{\boldsymbol{A}} + \boldsymbol{A}^i$
$\forall\, j, \boldsymbol{x}_j^0 = \sum_i \boldsymbol{T}_{iij}^0 + \boldsymbol{T}_{iji}^0 + \boldsymbol{T}_{jii}^0$;
$\boldsymbol{x}^0 = \boldsymbol{x}^0 / \|\boldsymbol{x}^0\|$;
for $k = 0$ to $m-2$ do
    $\boldsymbol{x}^{k+1} = \boldsymbol{T}^{k+1}(\boldsymbol{x}^k, \boldsymbol{x}^k, :) + \boldsymbol{T}^{k+1}(\boldsymbol{x}^k, :, \boldsymbol{x}^k) + \boldsymbol{T}^{k+1}(:, \boldsymbol{x}^k, \boldsymbol{x}^k)$;
    $\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k+1} / \|\boldsymbol{x}^{k+1}\|$;
return $\boldsymbol{x}^{m-1}$;
```

To get rid of the Independence Assumption, we slightly modify the algorithm (see Algorithm 2). In particular, we add more noise in every step as follows

- Get the input tensor $\boldsymbol{T} = \tau \cdot \boldsymbol{v}^{\otimes 3} + \boldsymbol{A}$;
- Draw a sequence of $\boldsymbol{A}^p \in \mathbb{R}^{n^{\otimes 3}}$ such that $\boldsymbol{A}_{ijk}^p \sim \mathcal{N}(0, m)$;
- Let $\boldsymbol{T}^p = \boldsymbol{T} - \overline{\boldsymbol{A}} + \boldsymbol{A}^p$ with $\overline{\boldsymbol{A}} = \frac{1}{m}\sum_{i=0}^{m-1} \boldsymbol{A}^i$, run Algorithm 2 by using $\boldsymbol{T}^p$ in the p-th iteration;

Intuitively, by adding more noise the new noise will overwhelm the original noise $\boldsymbol{A}$, and every time it looks like a fresh random noise. We prove this formally by the following lemma:

Lemma 2. *The sequence $\boldsymbol{T}^0, \cdots, \boldsymbol{T}^{m-1}$ has the same distribution as i.i.d Gaussians with mean $\tau \cdot \boldsymbol{v}^{\otimes 3}$ and variance $m\boldsymbol{J}$, where $\boldsymbol{J}$ is an all-one third-order tensor.*

The basic idea for this lemma is that for two multivariate Gaussians to have the same distribution, we only need to show that they have the same first and second moments. We defer the details to supplementary material.

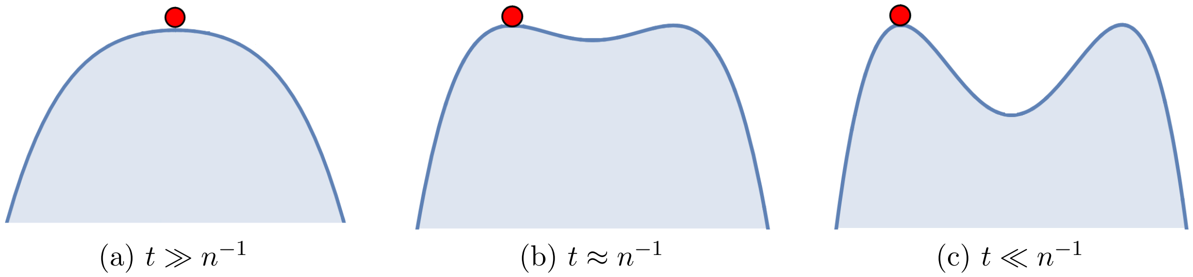
Using Lemma 2 we can create a sequence of G^t such that each $\boldsymbol{A}$ is redrawn independently and each element is according to $\mathcal{N}(0, m)$. Now we can prove the Independence Assumption:

Lemma 3 (Noise Injection)**.** *Let $\boldsymbol{T}^p = \boldsymbol{T} - \overline{\boldsymbol{A}} + \boldsymbol{A}^p$, then with high probability we have (1) $\|\delta(\boldsymbol{x}^p)\|_2 = \Theta(\sqrt{nm})\|\boldsymbol{x}^p\|_2^2$, (2) $\langle\delta(\boldsymbol{x}^p), \boldsymbol{v}\rangle = O(\sqrt{\log m})\|\boldsymbol{x}^p\|_2^2$, (3) $\|\boldsymbol{u}\|_2 = \Theta(n\sqrt{m})$, and (4) $\langle\boldsymbol{u}, \boldsymbol{v}\rangle = O(\sqrt{nm \log m})$.*

This Lemma is now true because by Lemma 2, we know the noise tensors $\boldsymbol{A}^p$ used in p-th step behave exactly the same as independent Gaussian tensors. This lemma then follows immediately from standard concentration inequalities. We defer the full proof to supplementary material.

Combining Lemma 3 and Theorem 3.1, we know Algorithm 2 solves the tensor PCA problem when $\tau = \tilde{\Omega}(n^{3/4})$.

4 Characterizing the Homotopy Path



(a) $t \gg n^{-1}$ (b) $t \approx n^{-1}$ (c) $t \ll n^{-1}$

Figure 1: Phase Transition for a 1-d function

In this section, we try to characterize how the smoothed objective function behaves for all values of t. We show in particular there is a phase transition: when t is large the function behaves very similarly to $t \to \infty$, and when t is small the function behaves very similarly to $t \to 0$.

4.1 Homotopy and Homotopy Path

Let us first describe how the homotopy method works in more detail. In the homotopy method, we start a maximizer with large amount of smoothing. Then we decrease the radius of smoothing t, and try to move the maximizer smoothly until we reach $t = 0$. We call the path taken by the maximizer the homotopy path.

Definition 2 (Homotopy Path)**.** A homotopy path is a function $\boldsymbol{x} : \mathcal{T} \to \mathcal{X}$, such that (1) $\lim_{t\to\infty} \boldsymbol{x}(t) = \boldsymbol{x}^\dagger$; (2) $\forall\ t \geq 0$, $\nabla g(\boldsymbol{x}(t), t) = 0$; (3) $\boldsymbol{x}(t)$ is continuous in t. Note that the gradient ∇ is w.r.t. to the first argument of g.

This path portrays one possible sequences of maximizers of g, starting with $x^\dagger$, as the parameter t decreases from ∞ to 0.

In practice, to search a homotopy path, one computes the initial point $\boldsymbol{x}^\dagger$ by analytically derivation or numerical approximation as $\arg\max_{\boldsymbol{x}} g(\boldsymbol{x}, t)$ for sufficiently large t. Then, the homotopy path $\boldsymbol{x}(t)$ is computed numerically until $t = 0$ as Algorithm 3.

4.2 Alternative Objective Function and Its Smoothing

For a constrained problem, it is not immediate how to compute the effective gradient and Hessian of $g(\boldsymbol{x}, t)$ under constraint $\|\boldsymbol{x}\|_2 = 1$. In this section, we will use the alternative objective function:

Algorithm 3: Homotopy Method

```
Input: $f : \mathcal{X} \to \mathbb{R}$, a sequence $t_0 > t_1 > \cdots > t_m = 0$.
Output: A (good) local maximizer of $f$.
$\boldsymbol{x}^0$ = global maximizer of $g(\boldsymbol{x}, t_0)$;
for $k = 1$ to $m$ do
    $\boldsymbol{x}^k$ = Local maximizer of $g(\boldsymbol{x}; t_k)$, initialized at $\boldsymbol{x}^{k-1}$.
return $\boldsymbol{x}^m$.
```

we modify $f(\boldsymbol{x})$ by adding the penalty term $-\frac{3\tau}{4} \cdot \|\boldsymbol{x}\|_2^4$:

$$f_r(\boldsymbol{x}) = \tau\langle \boldsymbol{v}, \boldsymbol{x}\rangle^3 + \boldsymbol{A}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) - \frac{3\tau}{4} \cdot \|\boldsymbol{x}\|_2^4$$

Thus we consider the following optimization,

$$\max \quad f_r(\boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) - \frac{3\tau}{4}\|\boldsymbol{x}\|_2^4. \tag{4}$$

If we fix the magnitude $\|\boldsymbol{x}\| = 1$, the function $f_r(\lambda\boldsymbol{x})$ is $\lambda^3\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) - \frac{3\tau}{4}\lambda^4$. The optimizer of this is an increasing function of $\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x})$. Therefore the maximizer of (4) is exactly in the same direction as the constrained problem (2). The $3\tau/4$ factor here is just to make sure the optimal solution has roughly unit norm; in practice we can choose any coefficient in front of $\|\boldsymbol{x}\|^4$ and the solution will only differ by scaling.

Notice that, if in the absence of noise tensor $\boldsymbol{A}$, then

$$\nabla f_r(\boldsymbol{x}) = 3\tau\langle \boldsymbol{v}, \boldsymbol{x}\rangle^2\boldsymbol{v} - \frac{3\tau}{4} \cdot 4\|\boldsymbol{x}\|_2^2\boldsymbol{x} = 0$$

To get the stationary point, we have

$$\boldsymbol{x} = \frac{3\tau}{4} \cdot \frac{\langle \boldsymbol{v}, \boldsymbol{x}\rangle^2}{\frac{3\tau}{4} \cdot \|\boldsymbol{x}\|_2^2}\boldsymbol{v} = \boldsymbol{v}$$

Therefore, the new function $f_r(\boldsymbol{x})$ is defined on $\mathbb{R}^n$ and the maximizer of $\mathbb{R}^n$ is close to $\boldsymbol{v}$. We also compute the smoothed version of this problem:

Lemma 4 (Smoothed Alternative Objective)**.** *The smoothed version of the alternative objective is*

$$\begin{aligned} g_r(\boldsymbol{x}, t) = &\tau\langle \boldsymbol{v}, \boldsymbol{x}\rangle^3 + t^2\langle 3\tau\boldsymbol{v} + \boldsymbol{u}, \boldsymbol{x}\rangle + \\ &\boldsymbol{A}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) - \frac{3\tau}{4}\left(\|\boldsymbol{x}\|_2^4 + 2t^2n\|\boldsymbol{x}\|_2^2 + t^4n^2\right) \end{aligned}$$

Its gradient and Hessian are equal to

$$\begin{aligned} \nabla g_r(\boldsymbol{x}, t) = &3\tau\langle \boldsymbol{v}, \boldsymbol{x}\rangle^2\boldsymbol{v} + t^2(3\tau\boldsymbol{v} + \boldsymbol{u}) \\ &+ \delta(\boldsymbol{x}) - 3\tau(\|\boldsymbol{x}\|_2^2\boldsymbol{x} + t^2n\boldsymbol{x}). \end{aligned} \tag{5}$$

and

$$\begin{aligned} \nabla^2 g_r(\boldsymbol{x}, t) = &- 3\tau((\|\boldsymbol{x}\|_2^2 + t^2n)\boldsymbol{I} - 2\langle \boldsymbol{v}, \boldsymbol{x}\rangle\boldsymbol{v}\boldsymbol{v}^T + 2\boldsymbol{x}\boldsymbol{x}^T) \\ &+ \boldsymbol{A}(\boldsymbol{x}, :, :) + \boldsymbol{A}(:, \boldsymbol{x}, :) + \boldsymbol{A}(:, :, \boldsymbol{x}). \end{aligned} \tag{6}$$

The proof of this Lemma is very similar to Lemma 1 and is deferred to supplementary material.

4.3 Phase Transition on the Homotopy Path

Notice that when $t \to \infty$, the dominating terms in $g_r(\boldsymbol{x}, t)$ are t^2 terms (the only t^4 term is a constant) form a quadratic function, so it has a unique global maximizer equal to $\frac{3\tau \boldsymbol{v}+\boldsymbol{u}}{n}$, we call this vector $\boldsymbol{x}^\dagger$. Notice that this vector has different norm compared to the $\boldsymbol{x}^\dagger$ in previous section.

Now we state the main result.

Theorem 4.1. *Assuming the Strong Independence Assumption (Assumption 2), when $\tau \geq n^{3/4} \log^2 n$, we know*

1. *When $t = \omega(n^{-1})$, there exists a local maximizer $\boldsymbol{x}^t$ of $g_r(\boldsymbol{x}, t)$ that is close to $\boldsymbol{x}^\dagger$.*
2. *When $t < n^{-1} \log^{-2} n$, we know there are two types of local maximizers $\boldsymbol{x}^t$:*
   - *$\|\boldsymbol{x}^t\|_2 = \Theta(1)$ and $\langle \boldsymbol{v}, \boldsymbol{x}^t \rangle = \Theta(1)$. This corresponds to a local maximizer near the true signal $\boldsymbol{v}$.*
   - *$\|\boldsymbol{x}^t\|_2 = \Theta(n^{-\frac{1}{4}}/\log n)$ and $\langle \boldsymbol{v}, \boldsymbol{x}^t \rangle = O(n^{-\frac{1}{2}}/\log^2 n)$. These local maximizers have poor correlation with the true signal.*
3. *When $t < n^{-1} \log^{-2} n$, for the point $\boldsymbol{x}^\dagger$, the top eigenvector of its Hessian $\nabla^2 g_r(\boldsymbol{x}, t)$ at $\boldsymbol{x}^\dagger$ is close to $\boldsymbol{v}$.*

Intuitively, this theorem shows that the global maximizer at infinite smoothing will gradually become a saddle point. From the saddle point we are very likely to follow the Hessian direction to actually converge to the good local maximizer near the signal. This is illustrated in Figure 1:

Figure 1(a) has large smoothing parameter, and the function has a unique local/global maximizer. Figure 1(b) has medium smoothing parameter, the original global maximizer now behaves like a local minimizer in one dimension, but it in general could be a saddle point in high dimensions. The Hessian at this point leads the direction of the homotopy path. In Figure 1(c) the smoothing is small and the algorithm should go to a different maximizer.

In the analysis of this section, we use a stronger version of the Independence Assumption.

Assumption 2. [Strong Independence Assumption] For all $\boldsymbol{x}$ on the homotopy path, we have (1) $\|\delta(\boldsymbol{x})\|_2 = \Theta(\sqrt{n})\|\boldsymbol{x}\|_2^2$, (2) $\langle \delta(\boldsymbol{x}), \boldsymbol{v} \rangle = \|\boldsymbol{x}\|_2^2$, (3) $\|\boldsymbol{u}\|_2 = \Theta(n)$, and (4) $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = O(\sqrt{n})$.

Intuitively, this assumes that the noise is not adversarially correlated with the signal $\boldsymbol{v}$ on the entire homotopy path. Although this assumption is strong and we cannot use noise injection to prove it, such assumptions are often used to get intuitions about optimization problems[8, 15, 7].

In order to characterize the homotopy path, we first provide the characterization of the local maximizers of function $g_r(\boldsymbol{x}, t)$ for different t. Notice that, after in $g_r(\boldsymbol{x}, \infty)$, the global maximizer is $\boldsymbol{x}^\dagger = \frac{\boldsymbol{v}}{n} + \frac{\boldsymbol{u}}{3\tau n}$ with norm $\Theta(\frac{1}{\tau})$ and correlation $\langle \boldsymbol{x}^\dagger, \boldsymbol{v} \rangle = \Theta(\frac{1}{n})$.

Lemma 5. *When $t = \omega(n^{-1})$, there exists a local maximizer $\boldsymbol{x}^t$ of $g_r(\boldsymbol{x}, t)$ such that $\|\boldsymbol{x}^t - \boldsymbol{x}^\dagger\|_2 = o(1)\|\boldsymbol{x}^\dagger\|_2$.*

From Lemma 5, the homotopy path remains at the starting point $\boldsymbol{x}^\dagger$ when $t = \Omega(n^{-1})$. However, when $t = \Theta(n^{-1})$, $\boldsymbol{x}^\dagger$ is no longer a local maximizer of $g_r(\boldsymbol{x}, t)$. We characterize what happens below the threshold in the following lemma:

Lemma 6. *When $t = n^{-1} \cdot \varepsilon(n)$, where $\varepsilon(n) = O(\log^{-2} n)$, the local maximizers (excluding saddle points) $\boldsymbol{x}^t$ of $g_r(\boldsymbol{x}, t)$ are of the following types:*

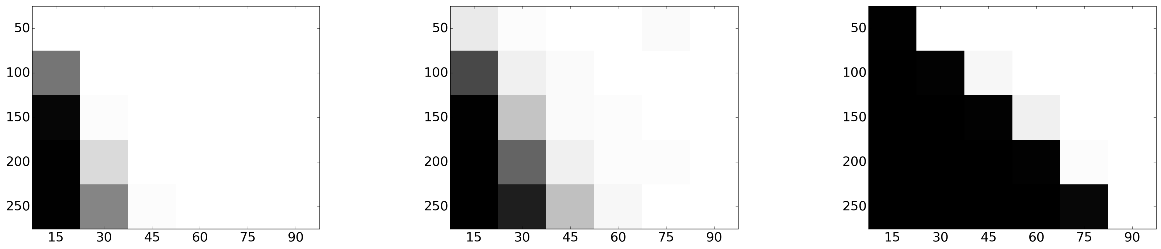
- $\|\boldsymbol{x}^t\|_2 = \Theta(1)$ *and* $\langle \boldsymbol{v}, \boldsymbol{x}^t \rangle = \Theta(1)$*;*
- $\|\boldsymbol{x}^t\|_2 = \Theta(n^{-\frac{1}{4}} \log^{-1} n)$ *and* $\langle \boldsymbol{v}, \boldsymbol{x}^t \rangle \leq O(n^{-\frac{1}{2}} \log^{-2} n)$*;*

Combining Lemma 5 and Lemma 6, we provide a clear phase transition on the local maximizers on $t = \tilde{\Theta}(n^{-1})$. To get intuition why a homotopy path converges to good maximizers instead of bad maximizers, we show the following lemma:

Lemma 7. *For $t = O(n^{-1} \cdot \varepsilon(n))$, where $\varepsilon = O(\log^{-2} n)$, the top eigenvector of $\nabla^2(g_r(\boldsymbol{x}^\dagger, t))$ is highly correlated to the signal $\boldsymbol{v}$.*

Therefore, for $t = n^{-1} \cdot \varepsilon(n)$, where $\varepsilon = O(\log^{-2} n)$, the Hessian clearly points to a direction highly correlated with the signal $\boldsymbol{v}$, which provides a good insight to design our simple but effective algorithm (Algorithm 1).

5 Experiments

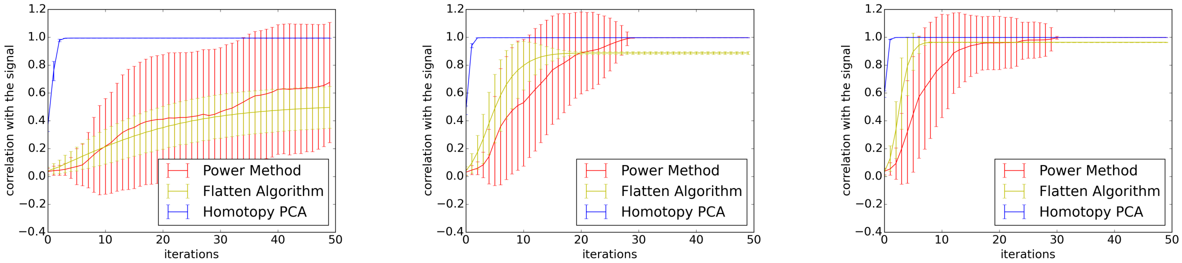


Homotopy PCA

Power Method

Flatten Algorithm

Figure 2: Success probabilities for the algorithms. y axis is n and x axis is τ. Black means fail.



$\alpha = 1.1$

$\alpha = 1.5$

$\alpha = 2$

Figure 3: Rate of Convergence. $\tau = \alpha n^{\frac{3}{4}}$, x axis is the number of iterations, y axis is the expected correlation with signal $\boldsymbol{v}$ (with variance represented as error bars)

For brevity we refer to our Tensor PCA with homotopy initialization method (Algorithm 1) as HomotopyPCA. We compare that with two other algorithms: the Flatten algorithm and the Power method. The Flatten algorithm was originally proposed by [20], where they show it works when $\tau = \Omega(n)$. [13] accelerated the Flatten algorithm to near-linear time, and improved the analysis to show it works when $\tau = \tilde{\Omega}(n^{3/4})$. The Power method is similar to our algorithm, except it does not use intuitions from homotopy, and initialize at a random vector. Note that there are other algorithms proposed in [13], however they are based on the Sum-of-Squares SDP hierarchy, and even the fastest version runs in time $O(n^5)$ (much worse than the $O(n^3)$ algorithms compared here).

We first compare how often these algorithms successfully find the signal vector v, given different values of τ and n. The results are in Figure 2, in which y-axis represents n and x-axis represents τ. We run 50 experiments for each values of (n, τ), and the grayness in each grid shows how frequent each algorithm succeeds: black stands for "always fail" and white stands "always succeed". For every algorithm, we say it fails if (1) when it converges, i.e., the result at two consecutive iterations are very close, the correlation with the signal $\boldsymbol{v}$ is less than 80%; (2) the number of iterations exceeds 100. In the experiments for Power Method, we observe there are many cases where situation (1) is true, although our new algorithms can always find the correct solution. In these cases the function indeed have a local maximizer. From Figure 2, our algorithm outperforms both Power Method and the Flatten algorithm in practice. This suggests the constant hiding in our algorithm is possibly smaller.

Next we compare the number of iterations to converge with $n = 500$ and $\tau = \alpha n^{\frac{3}{4}}$, where α varies in $[1.1, 1.5, 2]$. In Figure 3, the x-axis is the number of iterations, and the y axis is the correlation with the signal $\boldsymbol{v}$ (error bars shows the distribution from 50 independent runs). For all α, Homotopy PCA performs well — converges in less than 5 iterations and finds the signal $\boldsymbol{v}$. The Power Method converges to a result with good correlations with the signal $\boldsymbol{v}$, but has large variance because it sometimes gets trapped in local optima. As for the Flatten algorithm, the algorithm always converges. However, it takes more iterations compared to our algorithm. Also when α is small, the converged result has bad correlation with $\boldsymbol{v}$.

References

[1] Alekh Agarwal, Animashree Anandkumar, Prateek Jain, Praneeth Netrapalli, and Rashish Tandon. Learning sparsely used overcomplete dictionaries. In *Proceedings of The 27th Conference on Learning Theory*, pages 123–137, 2014.

[2] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[3] Animashree Anandkumar, Rong Ge, and Majid Janzamin. Learning overcomplete latent variable models through tensor methods. In *Proceedings of The 28th Conference on Learning Theory*, pages 36–112, 2015.

[4] Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *COLT*, pages 779–806, 2014.

[5] Quentin Berthet, Philippe Rigollet, et al. Optimal detection of sparse principal components in high dimension. *The Annals of Statistics*, 41(4):1780–1815, 2013.

[6] Alex Bloemendal and Bálint Virág. Limits of spiked random matrices i. *Probability Theory and Related Fields*, 156(3-4):795–825, 2013.

[7] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

[8] David L Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

[9] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory*, pages 797–842, 2015.

[10] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *arXiv preprint arXiv:1605.07272*, 2016.

[11] Elad Hazan, Kfir Y Levy, and Shai Shalev-Shwartz. On graduated optimization for stochastic non-convex problems. In *International Conference on Machine Learning (ICML)*, 2016.

[12] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

[13] Samuel B Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-square proofs. In *Proceedings of The 28th Conference on Learning Theory, COLT*, pages 3–6, 2015.

[14] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[15] Adel Javanmard and Andrea Montanari. State evolution for general approximate message passing algorithms, with applications to spatial coupling. *Information and Inference*, page iat004, 2013.

[16] Hossein Mobahi and John W Fisher III. On the link between gaussian homotopy continuation and convex envelopes. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 43–56. Springer, 2015.

[17] Hossein Mobahi and John W Fisher III. A theoretical analysis of optimization by gaussian continuation. In *AAAI*, pages 1205–1211. Citeseer, 2015.

[18] Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust pca. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

[19] Amelia Perry, Alexander S Wein, Afonso S Bandeira, and Ankur Moitra. Optimality and sub-optimality of pca for spiked random matrices and synchronization. *arXiv preprint arXiv:1609.05573*, 2016.

[20] Emile Richard and Andrea Montanari. A statistical model for tensor pca. In *Advances in Neural Information Processing Systems*, pages 2897–2905, 2014.

[21] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere. In *Sampling Theory and Applications (SampTA), 2015 International Conference on*, pages 407–410. IEEE, 2015.

[22] Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Arxiv*, 1602.06664, 2016.

[23] Xinyang Yi, Constantine Caramanis, and Sujay Sanghavi. Solving a mixture of many random linear equations by tensor decomposition and alternating minimization. *Arxiv*, abs/1608.05749, 2016.

A Omitted Proofs

A.1 Omitted Proof in Section 2

Lemma 8 (Lemma 1 restated)**.**

$$g(\boldsymbol{x},t) = \tau\langle \boldsymbol{v},\boldsymbol{x}\rangle^3 + t^2\langle 3\tau\boldsymbol{v}+\boldsymbol{u},\boldsymbol{x}\rangle + \boldsymbol{A}(\boldsymbol{x},\boldsymbol{x},\boldsymbol{x}),$$

where the vector $\boldsymbol{u}$ is defined by $\boldsymbol{u}_j = \sum_{i=1}^n(\boldsymbol{A}_{iij}+\boldsymbol{A}_{iji}+\boldsymbol{A}_{jii})$. Moreover, it is easy to compute vector $\boldsymbol{z} = 3\tau\boldsymbol{v}+\boldsymbol{u}$ given just the tensor $\boldsymbol{T}$, as $\forall j \quad \boldsymbol{z}_j = \sum_{i=1}^d(\boldsymbol{T}_{iij}+\boldsymbol{T}_{iji}+\boldsymbol{T}_{jii})$.

Proof. We can write $g(x,t)$ as an expectation

$$g(\boldsymbol{x},t) = \int_{\mathbb{R}^n} f(\boldsymbol{x}+\boldsymbol{y})k_t(\boldsymbol{y})dy = \mathbb{E}_{y\sim N(\mathbf{0},t^2\mathbf{I}_n)}[f(\boldsymbol{x}+\boldsymbol{y})] = \mathbb{E}_{y\sim N(\mathbf{0},\mathbf{I}_n)}[f(\boldsymbol{x}+t\boldsymbol{y})]$$

Since f is just a degree 3 polynomial, we can expand it and use the lower moments of Gaussian distributions:

$$\begin{aligned} g(\boldsymbol{x},t) &= \mathbb{E}[f(\boldsymbol{x}+t\boldsymbol{y})] \\ &= \mathbb{E}[\tau\langle\boldsymbol{v},(\boldsymbol{x}+\boldsymbol{t}y)\rangle^3 + \boldsymbol{A}(\boldsymbol{x}+t\boldsymbol{y},\boldsymbol{x}+t\boldsymbol{y},\boldsymbol{x}+t\boldsymbol{y})] \\ &= \tau\langle\boldsymbol{v},\boldsymbol{x}\rangle^3 + 3\tau t^2\langle\boldsymbol{v},\boldsymbol{x}\rangle\cdot\mathbb{E}[\langle\boldsymbol{v},\boldsymbol{y}\rangle^2] + \mathbb{E}[\boldsymbol{A}(\boldsymbol{x}+t\boldsymbol{y},\boldsymbol{x}+t\boldsymbol{y},\boldsymbol{x}+t\boldsymbol{y})] \\ &= \tau\langle\boldsymbol{v},\boldsymbol{x}\rangle^3 + 3\tau t^2\langle\boldsymbol{v},\boldsymbol{x}\rangle + t^2\sum_{i,j}(\boldsymbol{A}_{iij}+\boldsymbol{A}_{iji}+\boldsymbol{A}_{jii})\boldsymbol{x}_j + \boldsymbol{A}(\boldsymbol{x},\boldsymbol{x},\boldsymbol{x}) \end{aligned}$$

Therefore the first part of the lemma holds if we define $\boldsymbol{u}$ to be the vector $\boldsymbol{u}_j = \sum_i \boldsymbol{A}_{iij}+\boldsymbol{A}_{iji}+\boldsymbol{A}_{jii}$.

In order to compute the vector $3\tau\boldsymbol{v}+\boldsymbol{u}$, notice that the term $\langle 3\tau\boldsymbol{v}+\boldsymbol{u},\boldsymbol{x}\rangle$ is equal to

$$\langle 3\tau\boldsymbol{v}+\boldsymbol{u},\boldsymbol{x}\rangle = \mathbb{E}_{y\sim N(\mathbf{0},\mathbf{I}_n)}[\boldsymbol{T}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{y})+\boldsymbol{T}(\boldsymbol{y},\boldsymbol{x},\boldsymbol{y})+\boldsymbol{T}(\boldsymbol{y},\boldsymbol{y},\boldsymbol{x})].$$

This means

$$(3\tau\boldsymbol{v}+\boldsymbol{u})_j = \mathbb{E}_{y\sim N(\mathbf{0},\mathbf{I}_n)}[\boldsymbol{T}(\boldsymbol{e}_j,\boldsymbol{y},\boldsymbol{y})+\boldsymbol{T}(\boldsymbol{y},\boldsymbol{e}_j,\boldsymbol{y})+\boldsymbol{T}(\boldsymbol{y},\boldsymbol{y},\boldsymbol{e}_j)] = \sum_{i=1}^d(\boldsymbol{T}_{iij}+\boldsymbol{T}_{iji}+\boldsymbol{T}_{jii}).$$

□

Algorithm 4: Tensor PCA by Homotopy Initialization

```
**Input**: Tensor $\boldsymbol{T} = \tau \cdot v^{\otimes 3} + \boldsymbol{A}$;
**Output**: Approximation of $\boldsymbol{v}$;
$m = O(\log n / \log\log n)$;
$\forall\ j, \boldsymbol{x}^0_j = \sum_i \boldsymbol{T}_{iij} + \boldsymbol{T}_{iji} + \boldsymbol{T}_{jii}$;
$\boldsymbol{x}^0 = \boldsymbol{x}^0 / \|\boldsymbol{x}^0\|$;                    $//\boldsymbol{x}^0 = \boldsymbol{x}^\dagger$
**for** $k = 0$ *to* $m$ **do**
    $\boldsymbol{x}^{k+1} = \boldsymbol{T}(\boldsymbol{x}^k, \boldsymbol{x}^k, :) + \boldsymbol{T}(\boldsymbol{x}^k, :, \boldsymbol{x}^k) + \boldsymbol{T}(:, \boldsymbol{x}^k, \boldsymbol{x}^k)$;
    $\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k+1} / \|\boldsymbol{x}^{k+1}\|$;
**return** $\boldsymbol{x}^m$;
```

A.2 Omitted Proof in Section 3

Theorem A.1 (Theorem 3.1 restated)**.** *When $\tau = \tilde{\Omega}(n^{3/4})$, Algorithm 1 finds a vector $\boldsymbol{x}^m$ that is close to $\boldsymbol{v}$ in $O(\log n / \log\log n)$ iterations.*

Proof. We first show the initial maximizer $\boldsymbol{x}^0 = \boldsymbol{x}^\dagger$ already correlates with $\boldsymbol{v}$ well. We can bound $\|3\tau\boldsymbol{v} + \boldsymbol{u}\|_2$ from above and below by $\|\boldsymbol{u}\|_2 \pm \|3\tau\boldsymbol{v}\|_2$. However, the latter obeys $\Theta(n\sqrt{m}) \pm 3\Theta(n^{\frac{3}{4}} \cdot \log^c n)$, which simplifies to $\Theta(n\sqrt{m})$. Therefore, $\|3\tau\boldsymbol{v} + \boldsymbol{u}\|_2 = \Theta(n\sqrt{m})$. Thus, with high probability,

$$\begin{aligned}\langle \boldsymbol{x}^0, \boldsymbol{v} \rangle = \frac{3\tau + \langle \boldsymbol{u}, \boldsymbol{v} \rangle}{\|3\tau\boldsymbol{v} + \boldsymbol{u}\|_2} &= \frac{1}{\Theta(n\sqrt{m})}\Theta(n^{\frac{3}{4}} \cdot \log^c n) \\ &= \Theta(n^{-\frac{1}{4}} \cdot \log^c n / \sqrt{m})\end{aligned}$$

Let us first consider the first step of power method. Now consider the first iteration and let $\hat{\boldsymbol{x}}^1$ be the vector before normalization. Observe that $\hat{\boldsymbol{x}}^1 = 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0 \rangle^2 \boldsymbol{v} + \delta(\boldsymbol{x}^0)$. With high probability,

$$\begin{aligned}\langle \hat{\boldsymbol{x}}^1, \boldsymbol{v} \rangle &= 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0 \rangle^2 + O(\sqrt{\log m})\|\boldsymbol{x}^0\|_2^2 \\ &= 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0 \rangle^2 + O(\sqrt{\log m}),\end{aligned}$$

where we dropped $\|\boldsymbol{x}^0\|_2^2$ because it is equal to one (recall we normalize $\boldsymbol{x}^p$ in each iteration). In the first iteration, we have $\langle \hat{\boldsymbol{x}}^1, \boldsymbol{v} \rangle = \Theta(n^{\frac{1}{4}} \cdot \log^{3c} n / m)$. Moreover, $\|\hat{\boldsymbol{x}}^1\|_2$ can vary within $\|\delta(\boldsymbol{x}^0)\|_2 \pm \|3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0 \rangle^2 \boldsymbol{v}\|_2$, which simplifies to within $\Theta(\sqrt{nm}) \pm 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0 \rangle^2$. Again for the first iteration the first term dominates, $\|\hat{\boldsymbol{x}}^1\|_2 = \Theta(\sqrt{nm})$. Combining the bounds for the norm of $\hat{\boldsymbol{x}}^1$ and its correlation with $\boldsymbol{v}$, we know with high probability,

$$\langle \frac{\hat{\boldsymbol{x}}^1}{\|\hat{\boldsymbol{x}}^1\|}, \boldsymbol{v} \rangle = \Theta(n^{-\frac{1}{4}} \cdot \log^{3c} n / m^{\frac{3}{2}})$$

Therefore, when $\log^c n \gg m$, the correlation between $\boldsymbol{x}^1$ and $\boldsymbol{v}$ is larger than the correlation between $\boldsymbol{x}^0$ and $\boldsymbol{v}$. This shows the first step makes an improvement.

In order to show this for the future steps, we do induction over p. The induction hypothesis is for every p, either $\langle \boldsymbol{x}^p, \boldsymbol{v} \rangle \geq 0.9$ or

$$\langle \boldsymbol{x}^p, \boldsymbol{v} \rangle = \Theta(n^{-\frac{1}{4}} \log^{3^p c} n / m^{3^p - \frac{1}{2}})$$

Initially, for $p = 0$, this clearly holds.

Now assume this is true for p, in the next iteration, let $\hat{\boldsymbol{x}}^p$ be the vector before normalization. Similar as before we have $\hat{\boldsymbol{x}}^{p+1} = 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2 \boldsymbol{v} + \delta(\boldsymbol{x}^p)$, therefore with high probability we can compute its correlation with $\boldsymbol{v}$:

$$\begin{aligned}\langle \hat{\boldsymbol{x}}^{p+1}, \boldsymbol{v}\rangle &= 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2 \pm O(\sqrt{\log m})\\ &= \Theta(n^{\frac{1}{4}} \log^{3^{p+1}c} n / m^{3^{p+1}-1}).\end{aligned}$$

For the norm of $\hat{\boldsymbol{x}}^{p+1}$, notice that the first term $3\tau\langle \boldsymbol{v}$ has norm $3\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2$, and the second term $\delta(\boldsymbol{x}^p)$ has norm $\Theta(\sqrt{nm})$. Note that these two terms are almost orthogonal by Independence Assumption, therefore

$$\|\hat{\boldsymbol{x}}^{p+1}\|_2 = \Theta(\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2) + \Theta(\sqrt{nm})$$

If $3\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2 \gg C\sqrt{nm}$ for large enough constant C, then $\|\hat{\boldsymbol{x}}^{p+1}\|_2 \le (3+C')\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2)$, where C' is a constant that is smaller than 0.1 when C is large enough. Therefore in this case $\langle \frac{\hat{\boldsymbol{x}}^{p+1}}{\|\hat{\boldsymbol{x}}^{p+1}\|_2}, \boldsymbol{v}\rangle \ge 0.9$. Thus we successfully recover $\boldsymbol{v}$ in the next step.

Otherwise, we know $\|\hat{\boldsymbol{x}}^{p+1}\|_2 = \Theta(\sqrt{nm})$. Then,

$$\langle \frac{\hat{\boldsymbol{x}}^{p+1}}{\|\hat{\boldsymbol{x}}^{p+1}\|_2}, \boldsymbol{v}\rangle = \Theta(n^{-\frac{1}{4}} \cdot \log^{3^{p+1}c} n / m^{3^{p+1}-\frac{1}{2}})$$

If we select $c = 2$, after $m = O(\frac{\log n}{\log\log n})$ rounds, $\langle \boldsymbol{x}^m, \boldsymbol{v}\rangle$ will be large. □

Lemma 9 (Lemma 2 restated)**.** *The sequence $\boldsymbol{T}^0, \cdots, \boldsymbol{T}^{m-1}$ has the same distribution as i.i.d Gaussians with mean $\tau \cdot v^{\otimes 3}$ and variance $m\boldsymbol{J}$, where $\boldsymbol{J}$ is an all-one third-order tensor.*

Proof. In order to prove two multivariate Gaussians have the same distribution, we only need to show they have the same first and second moments.

First, we have $\mathbb{E}[G^p] = \tau \cdot v^{\otimes 3}$ for all p. Moreover, for all p, q, if $i \ne i'$, $j \ne j'$ or $k \ne k'$, we have

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{T}^p_{ijk}, \boldsymbol{T}^q_{i'j'k'}) &= \mathbb{E}[(\boldsymbol{T}^p_{ijk} - \tau\boldsymbol{v}_i\boldsymbol{v}_j\boldsymbol{v}_k)(\boldsymbol{T}^q_{i'j'k'} - \tau\boldsymbol{v}_{i'}\boldsymbol{v}_{j'}\boldsymbol{v}_{k'})]\\ &= \mathbb{E}[(\boldsymbol{A}^p_{ijk} - \overline{\boldsymbol{A}_{ijk}} + \boldsymbol{A}_{ijk})(\boldsymbol{A}^q_{i'j'k'} - \overline{\boldsymbol{A}_{i'j'k'}} + \boldsymbol{A}_{i'j'k'})]\\ &= \mathbb{E}[\boldsymbol{A}^p_{ijk} - \overline{\boldsymbol{A}_{ijk}} + \boldsymbol{A}_{ijk}]\mathbb{E}[\boldsymbol{A}^q_{i'j'k'} - \overline{\boldsymbol{A}_{i'j'k'}} + \boldsymbol{A}_{i'j'k'}]\\ &= 0\end{aligned}$$

Otherwise, if $p \ne q$,

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{T}^p_{ijk}, \boldsymbol{T}^q_{ijk}) &= \mathbb{E}[(\boldsymbol{T}^p_{ijk} - \tau\boldsymbol{v}_i\boldsymbol{v}_j\boldsymbol{v}_k)(\boldsymbol{T}^q_{ijk} - \tau\boldsymbol{v}_i\boldsymbol{v}_j\boldsymbol{v}_k)]\\ &= \mathbb{E}[(\boldsymbol{A}^p_{ijk} - \overline{\boldsymbol{A}_{ijk}} + \boldsymbol{A}_{ijk})(\boldsymbol{A}^q_{ijk} - \overline{\boldsymbol{A}_{ijk}} + \boldsymbol{A}_{ijk})]\\ &= -\frac{m-1}{m^2}\mathbb{E}[(\boldsymbol{A}^p_{ijk})^2 + (\boldsymbol{A}^q_{ijk})^2] + \sum_{l \ne p,q} \frac{1}{m^2}\mathbb{E}[(\boldsymbol{A}^l_{ijk})^2] + \mathbb{E}[\boldsymbol{A}^2_{ijk}]\\ &= -\frac{2(m-1)}{m^2} \cdot m + \frac{m-2}{m^2} \cdot m + 1 = 0\end{aligned}$$

Moreover, for the $p = q$,

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{T}^p_{ijk}, \boldsymbol{T}^p_{ijk}) &= \mathbb{E}[(\boldsymbol{T}^p_{ijk} - \tau \boldsymbol{v}_i\boldsymbol{v}_j\boldsymbol{v}_k)(\boldsymbol{T}^p_{ijk} - \tau \boldsymbol{v}_i\boldsymbol{v}_j\boldsymbol{v}_k)] \\ &= \mathbb{E}[(\boldsymbol{A}^p_{ijk} - \overline{\boldsymbol{A}_{ijk}} + \boldsymbol{A}_{ijk})(\boldsymbol{A}^p_{ijk} - \overline{\boldsymbol{A}_{ijk}} + \boldsymbol{A}_{ijk})] \\ &= -\frac{(m-1)^2}{m^2}\mathbb{E}[(\boldsymbol{A}^p_{ijk})^2] + \sum_{l \neq p} \frac{1}{m^2}\mathbb{E}[(\boldsymbol{A}^l_{ijk})^2] + \mathbb{E}[\boldsymbol{A}^2_{ijk}] \\ &= -\frac{(m-1)^2}{m^2} \cdot m + \frac{m-1}{m^2} \cdot m + 1 = m\end{aligned}$$

□

Lemma 10 (Lemma 3 restated)**.** *Let $\boldsymbol{T}^p = \boldsymbol{T} - \overline{\boldsymbol{A}} + \boldsymbol{A}^p$, then with high probability we have (1) $\|\delta(\boldsymbol{x}^p)\|_2 = \Theta(\sqrt{nm})\|x^p\|_2^2$, (2) $\langle \delta(\boldsymbol{x}^p), \boldsymbol{v}\rangle = O(\sqrt{\log m})\|\boldsymbol{x}^p\|_2^2$, (3) $\|\boldsymbol{u}\|_2 = \Theta(n\sqrt{m})$, and (4) $\langle \boldsymbol{u}, \boldsymbol{v}\rangle = O(\sqrt{nm\log m})$.*

Proof. Since by Lemma 2, we know the noise tensors $\boldsymbol{A}^p$ used in p-th step behave exactly the same as independent Gaussian tensors, we can prove this lemma by standard Gaussian concentration results. For terms like $\|u\|$ and $\|\delta(\boldsymbol{x}^p)\|$, we know the norm of a Gaussian random variable obeys the χ^2 distribution and is highly concentrated to its expectation. For terms like $\langle u, v\rangle$ and $\langle \delta(\boldsymbol{x}^p), v\rangle$, we know they are just Gaussian distributions and is always bounded by $O(\sigma\sqrt{\log m})$ with high probability. Therefore we only need to compute the expected norms of these vectors.

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{u}\|_2^2] &= \mathbb{E}[\sum_j(\sum_i \boldsymbol{A}_{iij} + \boldsymbol{A}_{iji} + \boldsymbol{A}_{jii})^2] \\ &= \mathbb{E}[\sum_j(\sum_{i\neq j} \boldsymbol{A}^2_{iij} + \boldsymbol{A}^2_{iji} + \boldsymbol{A}^2_{jii}) + 9\boldsymbol{A}^2_{iii}] \\ &= 3n(n-1)m + 9nm \\ &= \Theta(n^2 m)\end{aligned}$$

Therefore by standard concentration bounds we have $\|u\|_2 = \Theta(n\sqrt{m})$ with high probability.

$$\mathbb{E}[\langle \boldsymbol{u}, \boldsymbol{v}\rangle] = \mathbb{E}[\sum_j(\sum_i \boldsymbol{A}_{iij} + \boldsymbol{A}_{iji} + \boldsymbol{A}_{jii})\boldsymbol{v}_j] = 0$$

$$\begin{aligned}\mathbb{E}[\langle \boldsymbol{u}, \boldsymbol{v}\rangle^2] &= \mathbb{E}[\sum_j((\sum_i \boldsymbol{A}_{iij} + \boldsymbol{A}_{iji} + \boldsymbol{A}_{jii})\boldsymbol{v}_j)^2] \\ &= \mathbb{E}[\sum_j \boldsymbol{v}_j^2(9A^2_{iii} + \sum_{i\neq j} \boldsymbol{A}^2_{iij} + \boldsymbol{A}^2_{iji} + \boldsymbol{A}^2_{jii})] \\ &= 9m + 3(n-1)m \\ &= \Theta(nm)\end{aligned}$$

This means $\langle u, v\rangle$ is a Gaussian random variable with variance $\sigma^2 = \Theta(nm)$, therefore with high probability $|\langle u, v\rangle| \leq O(\sqrt{nm\log m})$.

Similarly we can compute the expected square norm of $\delta(\boldsymbol{x}^p)$ as below

$$\begin{aligned}\mathbb{E}[\|\delta(\boldsymbol{x}^p)\|_2^2] &= \Theta(1)\mathbb{E}[\|\boldsymbol{A}(\boldsymbol{x}^p, \boldsymbol{x}^p, :)\|_2^2] \\ &= \Theta(1)\mathbb{E}[\|\boldsymbol{A}(\boldsymbol{x}^p, \boldsymbol{x}^p, :)\|_2^2] \\ &= \Theta(1)\mathbb{E}[\sum_k(\sum_{i,j} \boldsymbol{A}_{ijk}\boldsymbol{x}_i^p\boldsymbol{x}_j^p)^2] \\ &= \Theta(1)\mathbb{E}[\sum_k(\sum_{i,j} \boldsymbol{A}_{ijk}^2(\boldsymbol{x}_i^p)^2(\boldsymbol{x}_j^p)^2)] \\ &= \Theta(1)nm\|x^p\|_2^4\end{aligned}$$

$$\mathbb{E}[\langle\delta(\boldsymbol{x}^p), \boldsymbol{v}\rangle] = \sum_{i,j,k}\mathbb{E}[\boldsymbol{A}_{ijk}(\boldsymbol{x}_i^p\boldsymbol{x}_j^p\boldsymbol{v}_k + \boldsymbol{x}_i^p\boldsymbol{v}_j\boldsymbol{x}_k^p + \boldsymbol{v}_i\boldsymbol{x}_j^p\boldsymbol{x}_k^p)] = 0$$

$$\begin{aligned}\mathbb{E}[\langle\delta(\boldsymbol{x}^p), \boldsymbol{v}\rangle^2] &= \sum_{i,j,k}\mathbb{E}[\boldsymbol{A}_{ijk}^2(\boldsymbol{x}_i^p\boldsymbol{x}_j^p\boldsymbol{v}_k + \boldsymbol{x}_i^p\boldsymbol{v}_j\boldsymbol{x}_k^p + \boldsymbol{v}_i\boldsymbol{x}_j^p\boldsymbol{x}_k^p)^2] \\ &= 3\sum_k \boldsymbol{v}_k^2\sum_{i,j}(\boldsymbol{x}_i^p)^2(\boldsymbol{x}_j^p)^2 + 6\sum_i(\boldsymbol{x}_i^p)^2\sum_{j,k}\boldsymbol{v}_i\boldsymbol{v}_j\boldsymbol{x}_j^p\boldsymbol{x}_k^p \\ &= 3\|\boldsymbol{x}^p\|_2^4 + 6\|\boldsymbol{x}^p\|_2^2\langle v, \boldsymbol{x}^p\rangle^2 \\ &\le 9\|\boldsymbol{x}^p\|_2^4\end{aligned}$$

The bounds on $\|\delta(\boldsymbol{x}^p)\|$ and $\langle\delta(\boldsymbol{x}^p), v\rangle$ follows immediately from these expectations. □

A.3 Omitted Proof in Section 4

Lemma 11 (Lemma 4 restated)**.** *The smoothed version of the alternative objective is*

$$\begin{aligned}g_r(\boldsymbol{x}, t) =& \tau\langle\boldsymbol{v}, \boldsymbol{x}\rangle^3 + t^2\langle 3\tau\boldsymbol{v} + \boldsymbol{u}, \boldsymbol{x}\rangle + \\ & \boldsymbol{A}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}) - \frac{3\tau}{4}\left(\|x\|_2^4 + 2t^2n\|x\|_2^2 + t^4n^2\right)\end{aligned}$$

Its gradient and Hessian are equal to

$$\nabla g_r(\boldsymbol{x}, t) = 3\tau\langle\boldsymbol{v}, \boldsymbol{x}\rangle^2\boldsymbol{v} + t^2(3\tau\boldsymbol{v} + \boldsymbol{u}) + \delta(\boldsymbol{x}) - 3\tau(\|\boldsymbol{x}\|_2^2\boldsymbol{x} + t^2n\boldsymbol{x}).$$

and

$$\nabla^2 g_r(\boldsymbol{x}, t) = -3\tau((\|\boldsymbol{x}\|_2^2 + t^2n)\boldsymbol{I} - 2\langle\boldsymbol{v}, \boldsymbol{x}\rangle\boldsymbol{v}\boldsymbol{v}^T + 2\boldsymbol{x}\boldsymbol{x}^T) + \boldsymbol{A}(\boldsymbol{x}, :, :) + \boldsymbol{A}(:, \boldsymbol{x}, :) + \boldsymbol{A}(:, :, \boldsymbol{x}).$$

Proof. Similar to Lemma 1, we can write the smoothing operation as an expectation. By linearity of expectation we know

$$g_r(\boldsymbol{x}, t) = g(\boldsymbol{x}, t) + \mathbb{E}[\|\boldsymbol{x} + t\boldsymbol{y}\|_2^4]$$

We can compute the new terms by the moments of Gaussians:

$$\mathbb{E}[\|\boldsymbol{x} + t\boldsymbol{y}\|_2^4] = \|x\|_2^4 + t^2(2n + 4)\|x\|_2^2 + t^4n^2 \approx \|x\|_2^4 + 2t^2n\|x\|_2^2 + t^4n^2$$

The equation for $g_r(\boldsymbol{x}, t)$ follows immediately, and since it is a polynomial it is easy to compute its gradient and Hessian. □

Lemma 12 (Lemma 5 restated)**.** *When $t = \omega(n^{-1})$, there exists a local maximizer $\boldsymbol{x}^t$ of $g_r(\boldsymbol{x}, t)$ such that $\|\boldsymbol{x}^t - \boldsymbol{x}^\dagger\|_2 = o(1)\|\boldsymbol{x}^\dagger\|_2$.*

Proof. We use the second order sufficient conditions, in order to prove there is a local maximizer we need to find a point with $\mathbf{0}$ gradient and negative definite Hessian.

From (5), we can derive the expression of stationary points,

$$\boldsymbol{x} = \frac{3\tau\langle \boldsymbol{v}, \boldsymbol{x}\rangle^2 \boldsymbol{v} + t^2(3\tau\boldsymbol{v} + \boldsymbol{u}) + \delta(\boldsymbol{x})}{3\tau(\|\boldsymbol{x}\|_2^2 + t^2 n)} \tag{7}$$

When $t = \omega(n^{-1})$, $\|\tau\langle \boldsymbol{v}, \boldsymbol{x}^\dagger\rangle^2 \boldsymbol{v}\|_2 = \omega(n^{-\frac{4}{5}})$, $\|t^2(3\tau\boldsymbol{v} + \boldsymbol{u})\|_2 = \Omega(n^{-1})$ and $\|\delta(\boldsymbol{x}^\dagger)\|_2 = \Theta(n^{-1})$. Therefore, $t^2(3\tau\boldsymbol{v} + \boldsymbol{u})$ dominates the numerator. Moreover, $t^2 n = \omega(n^{-1})$ and $\|\boldsymbol{x}^\dagger\|_2^2 = \Theta(n^{-\frac{3}{2}})$, and thus, $t^2 n$ dominates the denominator. Thus,

$$\begin{aligned}\frac{3\tau\langle \boldsymbol{v}, \boldsymbol{x}^\dagger\rangle^2 \boldsymbol{v} + t^2(3\tau\boldsymbol{v} + \boldsymbol{u}) + \delta(\boldsymbol{x}^\dagger)}{3\tau(\|\boldsymbol{x}^\dagger\|_2^2 + t^2 n)} &= \frac{(1 + o(1))t^2(3\tau\boldsymbol{v} + \boldsymbol{u})}{3\tau t^2 n} \\ &= (1 + o(1))\frac{\boldsymbol{v}}{n} + \frac{\boldsymbol{u}}{3\tau n} \\ &= (1 + o(1))\boldsymbol{x}^\dagger\end{aligned}$$

Moreover, applying similar scale analysis for the Hessian,

$$\nabla^2 g_r(\boldsymbol{x}^\dagger, t) = -3\tau t^2 n\boldsymbol{I} + \boldsymbol{A}(\boldsymbol{x}^\dagger, :, :) + \boldsymbol{A}(:, \boldsymbol{x}^\dagger, :) + \boldsymbol{A}(:, :, \boldsymbol{x}^\dagger)$$

while the spectral norm of $\boldsymbol{A}(\boldsymbol{x}, :, :) + \boldsymbol{A}(:, \boldsymbol{x}, :) + \boldsymbol{A}(:, :, \boldsymbol{x})$ is $\Theta(\sqrt{n})\|x^\dagger\|_2 = \Theta(n^{-\frac{1}{4}}) < \tau t^2 n$. Thus, with high probability, the Hessian is a positive semi-definite at the point $\boldsymbol{x}^\dagger$. □

Lemma 13 (Lemma 6 restated)**.** *When $t = n^{-1} \cdot \varepsilon(n)$, where $\varepsilon(n) = O(\log^{-2} n)$, the local maximizers (excluding saddle points) $\boldsymbol{x}^t$ of $g_r(\boldsymbol{x}, t)$ are of the following types:*

- *good maximizers: $\|\boldsymbol{x}^t\|_2^2 = \Theta(1)$ and $\langle \boldsymbol{v}, \boldsymbol{x}^t\rangle = \Theta(1)$;*
- *bad maximizers: $\|\boldsymbol{x}^t\|_2^2 = \Theta(n^{-\frac{1}{2}} \log^{-2} n)$ and $\langle \boldsymbol{v}, \boldsymbol{x}^t\rangle \leq O(n^{-\frac{1}{2}} \log^{-2} n)$;*

Proof. Now we use the second order necessary conditions. For all local maximizer, their gradient should be $\mathbf{0}$ and their Hessian should be negative semidefinite.

First, from (7), we can compute the inner product between $\boldsymbol{v}$ and $\boldsymbol{x}$:

$$\langle \boldsymbol{v}, \boldsymbol{x}\rangle = \frac{3\tau\langle \boldsymbol{v}, \boldsymbol{x}\rangle^2 + 3\tau t^2 + t^2\langle \boldsymbol{u}, \boldsymbol{v}\rangle + \langle \delta(\boldsymbol{x}), \boldsymbol{v}\rangle}{3\tau(\|\boldsymbol{x}\|_2^2 + t^2 n)} = \frac{3\tau\langle \boldsymbol{v}, \boldsymbol{x}\rangle^2 + 3\tau t^2 + O(1)\|\boldsymbol{x}\|_2^2}{3\tau(\|\boldsymbol{x}\|_2^2 + t^2 n)} \tag{8}$$

From (7), we can also compute the square of the norm of $\boldsymbol{x}$:

$$\|\boldsymbol{x}\|_2^2 = \frac{9\tau^2\langle \boldsymbol{v}, \boldsymbol{x}\rangle^4 + t^4\|3\tau\boldsymbol{v} + \boldsymbol{u}\|_2^2 + \|\delta(\boldsymbol{x})\|_2^2 + \eta(\boldsymbol{x})}{9\tau^2(\|\boldsymbol{x}\|_2^2 + t^2 n)^2}$$

where

$$\eta(\boldsymbol{x}) = 6\tau\langle \boldsymbol{v}, \boldsymbol{x}\rangle^2\langle \boldsymbol{v}, \delta(\boldsymbol{x})\rangle + 6\tau t^2\langle \boldsymbol{v}, \boldsymbol{x}\rangle^2(3\tau + \langle \boldsymbol{v}, \boldsymbol{u}\rangle) + 2t^2\langle 3\tau\boldsymbol{v} + \boldsymbol{u}, \delta(\boldsymbol{x})\rangle$$

is negligible in scale analysis. Therefore,

$$\|\boldsymbol{x}\|_2^2 = \frac{9\tau^2\langle \boldsymbol{v}, \boldsymbol{x}\rangle^4 + t^4\Theta(n^2) + \Theta(n)\|\boldsymbol{x}\|_2^4}{9\tau^2(\|\boldsymbol{x}\|_2^2 + t^2 n)^2} \tag{9}$$

We proceed the proof via a case analysis on the relative order between $\|\boldsymbol{x}\|_2^2$ and $t^2 n$.

$\|\boldsymbol{x}^t\|_2^2$ **dominates** t^2n**:**

First, recall that the Hessian at $\boldsymbol{x}^t$ must be a negative semidefinite. Therefore, $\tau\|\boldsymbol{x}^t\|_2^2$ must be larger than the spectral norm of $\boldsymbol{A}(\boldsymbol{x}^t,:,:)+\boldsymbol{A}(:,\boldsymbol{x}^t,:)+\boldsymbol{A}(:,:,\boldsymbol{x}^t)$. That is, $\tau\|\boldsymbol{x}^t\|_2^2 > \Theta(\sqrt{n})\|\boldsymbol{x}^t\|_2$, equivalent to $\|\boldsymbol{x}^t\|_2 = \Omega(n^{-\frac{1}{4}}\log^{-1} n)$. As a result, $\Theta(n)\|\boldsymbol{x}^t\|_2^4$ dominates $t^4\Theta(n^2)$ in the nominator of (9). Henceforth, we have

$$\|\boldsymbol{x}^t\|_2^2 = \frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^4}{\|\boldsymbol{x}^t\|_2^4} + \Theta(n^{-\frac{1}{2}}\log^{-2} n)$$

(1) If $\|\boldsymbol{x}^t\|_2^2 = \Theta(n^{-\frac{1}{2}}\log^{-2} n)$, plug it into (8), we have

$$\langle \boldsymbol{v},\boldsymbol{x}^t\rangle = \frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2}{\|\boldsymbol{x}^t\|_2^2} + \frac{t^2}{\|\boldsymbol{x}^t\|_2^2} + \frac{O(1)}{\tau}$$

Therefore, the largest possible $\langle \boldsymbol{v},\boldsymbol{x}^t\rangle$ is $\Theta(n^{-\frac{1}{2}}\log^{-2} n)$.

(2) If $\|\boldsymbol{x}^t\|_2^3 = \langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2$, plug it into (8):

$$\langle \boldsymbol{v},\boldsymbol{x}^t\rangle = \|\boldsymbol{x}^t\|_2 + \frac{t^2}{\|\boldsymbol{x}^t\|_2^2} + \frac{O(1)}{\tau} = \|\boldsymbol{x}^t\|_2$$

Thus, we can conclude $\|\boldsymbol{x}^t\|_2 = \langle \boldsymbol{v},\boldsymbol{x}^t\rangle = \Theta(1)$.

t^2n **dominates** $\|\boldsymbol{x}^t\|_2^2$**:**

We will show this case cannot happen.

Recall that the Hessian at $\boldsymbol{x}^t$ must be a negative semidefinite. Therefore, $\tau t^2 n$ must be larger than the spectral norm of $\boldsymbol{A}(\boldsymbol{x}^t,:,:)+\boldsymbol{A}(:,\boldsymbol{x}^t,:)+\boldsymbol{A}(:,:,\boldsymbol{x}^t)$. That is, $\tau t^2 n > \Theta(\sqrt{n})\|\boldsymbol{x}^t\|_2$, equivalent to $\|\boldsymbol{x}^t\|_2 = t^2\tau O(\sqrt{n})$. As a result, $3\tau t^2$ dominates $O(1)\|\boldsymbol{x}^t\|_2^2$ in the nominator of (8). Henceforth, we have

$$\langle \boldsymbol{v},\boldsymbol{x}^t\rangle = \frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2}{t^2 n} + \frac{1}{n}$$

Notice that if $\frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2}{t^2n}$ dominates n^{-1}, then $\langle \boldsymbol{v},\boldsymbol{x}^t\rangle = t^2 n$, implying $\frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2}{t^2n} = t^2 n = \frac{\varepsilon^2(n)}{n} \ll n^{-1}$. Thus, $\langle \boldsymbol{v},\boldsymbol{x}^t\rangle$ can only be $\Theta(n^{-1})$.

Moreover, notice that $t^4\Theta(n^2)$ dominates $\Theta(n)\|\boldsymbol{x}^t\|_2^4 = t^8\tau^4 O(n^3)$. Therefore, from (9),

$$\|\boldsymbol{x}^t\|_2^2 = \frac{1}{t^4}\Theta(n^{-6}) + \Theta(\frac{1}{\tau^2}) \Rightarrow \|\boldsymbol{x}^t\|_2 = \Omega(n^{-\frac{3}{4}}\log^{-1} n)$$

This contradicts with $\|\boldsymbol{x}^t\|_2 = O(n^{-\frac{3}{4}}\log^{-3c} n)$. □

Lemma 14 (Lemma 7 restated)**.** *For $t = O(n^{-1}\cdot\varepsilon(n))$, where $\varepsilon = O(\log^{-2} n)$, the top eigenvector of $\nabla^2(g_r(\boldsymbol{x}^\dagger,t))$ is highly correlated to the signal $\boldsymbol{v}$.*

Proof. Recall (6),

$$\nabla^2 g_r(\boldsymbol{x}^\dagger,t) = -3\tau((\|\boldsymbol{x}^\dagger\|_2^2 + t^2 n)\boldsymbol{I} - 2\langle \boldsymbol{v},\boldsymbol{x}^\dagger\rangle \boldsymbol{v}\boldsymbol{v}^T + 2\boldsymbol{x}^\dagger {\boldsymbol{x}^\dagger}^T) + \boldsymbol{A}(\boldsymbol{x}^\dagger,:,:) + \boldsymbol{A}(:,\boldsymbol{x}^\dagger,:) + \boldsymbol{A}(:,:,\boldsymbol{x}^\dagger)$$

and $\boldsymbol{x}^\dagger = \frac{\boldsymbol{v}}{n} + \frac{\boldsymbol{u}}{3\tau n}$ with norm $\Theta(\frac{1}{\tau})$ and correlation $\langle \boldsymbol{x}^\dagger,\boldsymbol{v}\rangle = \Theta(\frac{1}{n})$. Therefore, we have $\|\boldsymbol{x}^\dagger\|_2^2 + t^2 n = O(n^{-1}\cdot\varepsilon^2(n))$ and the spectral norm of $\boldsymbol{A}(\boldsymbol{x}^\dagger,:,:)+\boldsymbol{A}(:,\boldsymbol{x}^\dagger,:)+\boldsymbol{A}(:,:,\boldsymbol{x}^\dagger)$ is $\Theta(n^{-\frac{1}{4}}\log^{-1} n)$. Thus, $6\tau\langle \boldsymbol{v},\boldsymbol{x}^\dagger\rangle = \Theta(n^{-\frac{1}{4}}\log n)$ dominates the $\nabla^2 g_r(\boldsymbol{x}^\dagger,t)$, which provides the top eigenvector $\boldsymbol{v}$. □