Proceedings of Machine Learning Research vol 65:1–26, 2017

Homotopy Analysis for Tensor PCA

Anima Anandkumar A.ANANDKUMAR@UCI.EDU
University of California, Irvine

Yuan Deng ERICDY@CS.DUKE.EDU
Duke University

Rong Ge RONGGE@CS.DUKE.EDU
Duke University

Hossein Mobahi HMOBAHI@CSAIL.MIT.EDU
Google Research

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 us 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. Unfortunately, theoretical guarantees for nonconvex optimization have been mostly negative, and the problems are computationally hard in the worst case. Nevertheless, simple local-search algorithms such as stochastic gradient descent have enjoyed great empirical success in areas such as deep learning. As such, recent research efforts have attempted to bridge this gap between theory and practice.

For example, one property that can guarantee the success of local search methods over nonconvex functions is when all local minima are also the global minima. Interestingly, it has been recently proven that many well known nonconvex problems do have this property, 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 (Ge et al., 2016), orthogonal tensor decomposition (Anandkumar et al., 2014; Ge et al., 2015), phase retrieval (Sun et al., 2016), complete dictionary learning (Sun et al., 2015), 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.

The above property, while very helpful, imposes a strong assumption on the nonconvex problem. A less restrictive requirement for the success of local search methods is the ability to initialize local search in the basin of attraction of the global optimum using another polynomial-time algorithm. This approach does not require all the local optima to be of good quality, and thus can cover a broader set of problems. Efficient initialization strategies have recently been developed for many nonconvex problems such as overcomplete dictionary learning (Arora et al., 2014; Agarwal et al., 2014), tensor decomposition (Anandkumar et al., 2015), robust PCA (Netrapalli et al., 2014), mixed linear regression (Yi et al., 2016) and so on.

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

1.1. Homotopy Method

The homotopy method is a general and a problem independent technique for tackling nonconvex problems. It starts from an objective function that is efficient to optimize (e.g. convex function), and progressively transforms it to the required objective (Mobahi and Fisher III, 2015b). Throughout this progression, the solution of each intermediate objective is used to initialize a local search on the next one. A particular 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 (Mobahi and Fisher III, 2015a; Hazan et al., 2016). However, the assumptions in these results are either too restrictive (Mobahi and Fisher III, 2015a) or extremely difficult to check (Hazan et al., 2016). 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 drastically simplified without sacrificing the theoretical guarantees. Specifically, 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 of the homotopy, and use its solution to initialize local search on the original objective. Thus, we show that the homotopy method can serve as a problem independent principle for obtaining a smart initialization which is then employed in 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 a broader set of nonconvex problems.

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	$\tilde{\Omega}(n^{3/4})$	$> \Omega(n^6)$	$> \Omega(n^6)$
Recover 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 Richard and Montanari (2014). sum-of-squares, Recover and Certify, and flattened tensor were analyzed in Hopkins et al. (2015).

1.2. Tensor PCA

Tensor PCA problem is an extension of the matrix PCA. The statistical model for tensor PCA was first introduced by Richard and Montanari (2014). 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 $i,j,k \in [n]$.

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

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 analysis (PCA), to recover the unknown signal has been studied under various noise models (Perry et al., 2016; Bloemendal and Virág, 2013) and sparsity assumptions on the signal vector (Berthet et al., 2013).

Previous algorithms for tensor PCA belong to two classes: local search methods such as tensor power iterations (Richard and Montanari, 2014), and global methods such as sum of squares (Hopkins et al., 2015). 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. Hopkins et al. (2015) also showed a lower-bound 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.

1.3. Contributions

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 introducing a specific initialization and injecting appropriate random noise in each iteration. This runs almost in linear time; see Table 1 for more details.

Theorem 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).

Our algorithm is inspired by the homotopy framework. In particular, we establish a phase transition along the homotopy path.

Theorem 2 (informal) *Under a plausible independence conjecture, 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. In the proof we also design a new noise injection procedure that breaks the dependency between the steps. This allows for simpler analysis and our algorithm does not rely on the independence conjecture. We discuss this in more detail in Section 3.1.

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 associated objective function. Then we show how to compute the smoothed versions of these objective functions.

2.1. Tensors and Polynomials

Tensors are higher dimensional generalization of matrices. In this paper we focus on 3rd order tensors, which correspond to a 3 dimensional arrays. Given a vector $\boldsymbol{v} \in \mathbb{R}^n$, similar to 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 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 ($\boldsymbol{A}_{ijk}$ is not necessarily equal to $\boldsymbol{A}_{jik}$ or other permutations), we also define the symmetric operator

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

2.2. Objective Functions for Tensor PCA

We first define the tensor PCA problem formally.

Definition 3 (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 $\langle \boldsymbol{x}, \boldsymbol{v} \rangle \geq 0.8$).*

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.

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

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. Richard and Montanari (2014) noticed that:

Theorem 4 *When $\tau \geq C\sqrt{n}$ for large constant C, the global optimum of (2) is close to the signal $\boldsymbol{v}$.*

Unfortunately, solving this optimization problem is NP-hard in the worst-case (Hillar and Lim, 2013). Currently, 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 ($\Omega(n^{3/4})$).

2.3. Gaussian Smoothing for the Objective Function

Guaranteed homotopy methods rely on smoothing the objective function by the Gaussian kernel (Mobahi and Fisher III, 2015b,a). 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 $\mathcal{N}(\mathbf{0}, t^2 \boldsymbol{I}_n)$, satisfying

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

It is known that convolution of polynomials with the Gaussian kernel has a closed form expression (Mobahi, 2016). In particular, the objective function of the Tensor PCA has the following smoothed form.

Lemma 5 (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 $\forall j, \boldsymbol{z}_j = \sum_{i=1}^n(\boldsymbol{T}_{iij} + \boldsymbol{T}_{iji} + \boldsymbol{T}_{jii})$.

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 A.1

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 5, we can compute vector $\boldsymbol{z}$ $\boldsymbol{z}_j = \sum_{i=1}^n \boldsymbol{T}_{iij} + \boldsymbol{T}_{iji} + \boldsymbol{T}_{jii}$, and we know $\boldsymbol{z} = 3\tau\boldsymbol{v} + \boldsymbol{u}$. Therefore we know $\boldsymbol{x}^\dagger = \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_2}$ can be computed just from the tensor. 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 use the following *independence* condition, which states that the "random"-looking vectors $\boldsymbol{u}$ and $\delta(\boldsymbol{x}^p) = \boldsymbol{A}(\boldsymbol{x}^p, \boldsymbol{x}^p, :) + \boldsymbol{A}(\boldsymbol{x}^p, :, \boldsymbol{x}^p) + \boldsymbol{A}(:, \boldsymbol{x}^p, \boldsymbol{x}^p)$ indeed have some properties satisfied by random vectors:

Condition 6 *[Independence Condition] The norm and correlation with $\boldsymbol{v}$ for the vectors $\boldsymbol{u}$ and $\delta(\boldsymbol{x}^p)$ are not far from expectation. More precisely: (1) $\|\boldsymbol{u}\|_2 = O(n\sqrt{m})$ and $|\langle \boldsymbol{u}, \boldsymbol{v}\rangle| = O(\sqrt{nm\log n})$; (2) for the sequence computed by Algorithm 1, $\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^m$, $\forall 0 \le p \le m$, $\|\delta(\boldsymbol{x}^p)\|_2 = O(\sqrt{nm})\|\boldsymbol{x}^p\|_2^2$ and $|\langle\delta(\boldsymbol{x}^p), \boldsymbol{v}\rangle| = O(\sqrt{m\log n})\|\boldsymbol{x}^p\|_2^2$.*

Note that if in every step of the algorithm, the noise tensor $\boldsymbol{A}$ is *resampled* to be a fresh random tensor, independent of the previous step $\boldsymbol{x}^p$, then $\delta(\boldsymbol{x}^p)$ is just a random Gaussian vector. In this case the condition is trivially satisfied. Of course, in reality $\boldsymbol{x}^i$'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. The extra dependency on m in Condition 6 comes from noise injection procedure and will be more clear in Section 3.1. We will first show the correctness of the algorithm assuming independence condition here, and in Section 3.1 we discuss the noise injection procedure and prove the independence condition.

Theorem 7 *When $\tau \geq Cn^{3/4} \log n$ for a large enough constant C, under the Independence Condition (Condition 6), Algorithm 1 finds a vector $\boldsymbol{x}^m$ such that $\langle \boldsymbol{x}^m, \boldsymbol{v} \rangle \geq 0.8$ in $O(\log \log n)$ iterations.*

Algorithm 1: Tensor PCA by Homotopy Initialization

```
Input: Tensor $\boldsymbol{T} = \tau \cdot \boldsymbol{v}^{\otimes 3} + \boldsymbol{A}$;
Output: Approximation of $\boldsymbol{v}$;
$m = O(\log \log n)$;
$\forall\, j, \boldsymbol{x}_j^0 = \sum_i \boldsymbol{T}_{iij} + \boldsymbol{T}_{iji} + \boldsymbol{T}_{jii}$;
$\boldsymbol{x}^0 = \boldsymbol{x}^0 / \|\boldsymbol{x}^0\|$;                                  // Now $\boldsymbol{x}^0 = \boldsymbol{x}^\dagger$
for $k = 0$ to $m$ do
    $\boldsymbol{x}^{k+1} = \boldsymbol{T}(\boldsymbol{x}^k, \boldsymbol{x}^k, :) + \boldsymbol{T}(\boldsymbol{x}^k, :, \boldsymbol{x}^k) + \boldsymbol{T}(:, \boldsymbol{x}^k, \boldsymbol{x}^k)$;
    $\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k+1} / \|\boldsymbol{x}^{k+1}\|$;
end
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 Condition. 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 and we have $\hat{\boldsymbol{x}}^{k+1} = 3\tau \langle \boldsymbol{v}, \boldsymbol{x}^k \rangle^2 \boldsymbol{v} + \delta(\boldsymbol{x}^k)$. Notice that the first term is in the direction $\boldsymbol{v}$, and the Independence Condition 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 Appendix A.2.

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 $\boldsymbol{v}$;
$m = O(\log \log n)$;
Sample $\boldsymbol{B}^0, \boldsymbol{B}^1, ..., \boldsymbol{B}^{m-1} \in \mathbb{R}^{n \times n \times n}$ whose entries are $\mathcal{N}(0, m)$.
Let $\overline{\boldsymbol{B}} = \frac{1}{m} \sum_{p=0}^{m-1} \boldsymbol{B}^p$.
Let $\boldsymbol{T}^p = \boldsymbol{T} - \bar{\boldsymbol{B}} + \boldsymbol{B}^p$
$\forall\, j, \boldsymbol{x}_j^0 = \sum_i \boldsymbol{T}_{iij}^0 + \boldsymbol{T}_{iji}^0 + \boldsymbol{T}_{jii}^0$;
$\boldsymbol{x}^0 = \boldsymbol{x}^0 / \|\boldsymbol{x}^0\|$;
for $k = 0$ to $m - 2$ do
    $\boldsymbol{x}^{k+1} = \boldsymbol{T}^{k+1}(\boldsymbol{x}^k, \boldsymbol{x}^k, :) + \boldsymbol{T}^{k+1}(\boldsymbol{x}^k, :, \boldsymbol{x}^k) + \boldsymbol{T}^{k+1}(:, \boldsymbol{x}^k, \boldsymbol{x}^k)$;
    $\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k+1} / \|\boldsymbol{x}^{k+1}\|$;
end
return $\boldsymbol{x}^{m-1}$;
```

In order to prove the Independence Condition, 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 $\boldsymbol{B}^p \in \mathbb{R}^{n \otimes 3}$ such that $\boldsymbol{B}^p_{ijk} \sim \mathcal{N}(0, m)$;
- Let $\boldsymbol{T}^p = \boldsymbol{T} - \overline{\boldsymbol{B}} + \boldsymbol{B}^p$ with $\overline{\boldsymbol{B}} = \frac{1}{m}\sum_{p=0}^{m-1} \boldsymbol{B}^p$, 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 8 *Let the sequence $\boldsymbol{T}^0, \cdots, \boldsymbol{T}^{m-1}$ be generated according to Algorithm 2. Let $\boldsymbol{Q}^i = \tau \boldsymbol{v}^{\otimes 3} + \boldsymbol{C}^i$, where $\boldsymbol{C}^i$'s are tensors with independent Gaussian entries. Each entry in $\boldsymbol{C}^i$ is distributed as $N(0, m)$. The two sets of variables $\{\boldsymbol{T}^i\}$ and $\{\boldsymbol{Q}^i\}$ has the same distribution.*

This Lemma states that after our noise injection procedure, the tensors $\boldsymbol{T}^0, ..., \boldsymbol{T}^{m-1}$ look *exactly the same* as tensors where the noise $\boldsymbol{A}$ is sampled independently. 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 Appendix A.2.

Using Lemma 8 we can create a sequence of T^p such that its noise tensor $\boldsymbol{A}^p = \boldsymbol{A} - \overline{\boldsymbol{B}} + \boldsymbol{B}^p$ is redrawn independently and each element is according to $\mathcal{N}(0, m)$. Now, because each $\boldsymbol{T}^i$ behave as if it is drawn independently, we can prove the Independence Condition:

Lemma 9 (Noise Injection) *Let $\boldsymbol{T}^p$ be generated according to Algorithm 2 and $\boldsymbol{A}^p = \boldsymbol{T}^p - \tau \boldsymbol{v}^{\otimes 3}$. Let $\boldsymbol{u}^0$ be a vector such that $\boldsymbol{u}^0_j = \sum_i \boldsymbol{A}^0_{iij} + \boldsymbol{A}^0_{iji} + \boldsymbol{A}^0_{jii}$, and $\delta^p(\boldsymbol{x}^p) = \boldsymbol{A}^p(\boldsymbol{x}^p, \boldsymbol{x}^p, :) + \boldsymbol{A}^p(\boldsymbol{x}^p, :, \boldsymbol{x}^p) + \boldsymbol{A}^p(:, \boldsymbol{x}^p, \boldsymbol{x}^p)$. With high probability*[1]*, (1) $\|\boldsymbol{u}^0\|_2 = \Theta(n\sqrt{m})$ and $|\langle \boldsymbol{u}^0, \boldsymbol{v}\rangle| = O(\sqrt{nm \log n})$; (2) for the sequence computed by Algorithm 2, $\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^{m-1}$, $\forall\, 0 \le p \le m-1$, $\|\delta^p(\boldsymbol{x}^p)\|_2 = \Theta(\sqrt{nm})\|\boldsymbol{x}^p\|_2^2$ and $|\langle \delta^p(\boldsymbol{x}^p), \boldsymbol{v}\rangle| = O(\sqrt{m \log n})\|\boldsymbol{x}^p\|_2^2$. As a result Condition 6 is satisfied.*

This Lemma is now true because by Lemma 8, we know the noise tensors $\boldsymbol{A}^p$ is independent of $\boldsymbol{A}^0, ..., \boldsymbol{A}^{p-1}$. As a result $\boldsymbol{A}^p$ is *independent* of $\boldsymbol{x}^p$! This lemma then follows immediately from standard concentration inequalities. We defer the full proof to Appendix A.2.

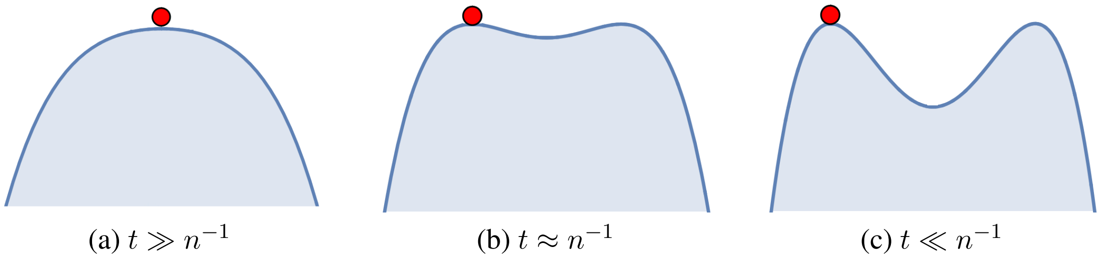
The noise injection technique is mostly a technicality that we need in order to make sure different steps are independent. This is standard in analyzing nonconvex optimization algorithms. As an example, previous works on alternating minimization for matrix completion (Jain et al., 2013) relied on the 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 main usage of the noise injection is to get rid of the dependence of the noise matrix between different iterations. Moreover, this technique is designed to simplify the proof and rarely used in the real applications. In practice, an algorithm without noise injection, like Algorithm 1, usually performs well enough.

Combining Lemma 9 and Theorem 7, we know Algorithm 2 solves the tensor PCA problem when $\tau \ge C n^{3/4} \log n$.

[1] Throughout this paper by "with high probability" we mean the probability is at least $1 - 1/n^C$ for a large constant C.

Remark 10 (Estimation of the variance in practice) *In the above analysis, we assume the variance of entries of $\boldsymbol{A}$ is 1. In practice, we can estimate the variance σ^2 of entries of $\boldsymbol{A}$ from $\boldsymbol{T}$ by computing its Frobenius norm. Note that when τ is large, the simple power method already performs well. The interesting case is when τ is small, say $\tau < n$. In this case, the square of the Frobenius norm of $\tau \boldsymbol{v}^{\otimes 3} = \tau^2$ while the square of the Frobenius norm of the noise matrix $\boldsymbol{A}$ in expectation is $\sigma^2 n^3$ with variance $\sigma^2 O(n^3)$. Therefore, we can get a good estimation of σ^2 by computing the square of the Frobenius norm of $\boldsymbol{A}$ divided by n^3.*

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

This section analyzes the behavior of the smoothed objective function g as t varies. Under a plausible conjecture, we prove that a phase transition occurs: when t is large $g(\boldsymbol{x}, t)$ behaves very similarly to $g(\boldsymbol{x}, \infty)$ and when t is small $g(\boldsymbol{x}, t)$ behaves very similarly to $g(\boldsymbol{x}, 0)$. This motivates the algorithms in the previous section, as the phase transition suggests the most important regimes are very large t and $t = 0$.

In this section we first describe how the homotopy method works in more details. Then we present an alternative objective function of Tensor PCA and derive its smoothed version. Finally, we prove that when $t \gg n^{-1}$, the smoothed function retains its maximizer around $\boldsymbol{x}^\dagger$. However, when $t \ll n^{-1}$, the configuration of critical points change, with only one of the critical points being close to the solution $\boldsymbol{v}$. Importantly, we can find our way from the vicinity of $\boldsymbol{x}^\dagger$ toward this critical point via the dominant curvature direction of the function.

4.1. Homotopy

In the homotopy method, we start from the maximizer of the function with large amount of smoothing $t \to \infty$. We earlier denoted this maximizer as $\boldsymbol{x}^\dagger$. Then we continuously decrease the amount of smoothing t, while following the maximizer throughout this process, until reaching $t = 0$. We call the path taken by the maximizer the homotopy path. It is formally defined as follows.

Definition 11 (Homotopy Path) *A homotopy path $\boldsymbol{x}(t)$ is a continuous function $\boldsymbol{x} : \mathcal{T} \to \mathcal{X}$ satisfying $\lim_{t\to\infty} \boldsymbol{x}(t) = \boldsymbol{x}^\dagger$ and $\forall\, t \geq 0$, $\nabla g(\boldsymbol{x}(t), t) = 0$, where the gradient ∇ is w.r.t. to the first argument of g.*

In practice, to search a homotopy path, one computes the initial point $\boldsymbol{x}^\dagger$ by analytical derivation or numerical approximation as $\arg\max_{\boldsymbol{x}} g(\boldsymbol{x}, t)$ and then successively minimizes the smoothed functions over a finite sequence of decreasing numbers t_0 to t_m, where t_0 is sufficiently large, and $t_m = 0$. The resulted procedure is listed in Algorithm 3.

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 $f$.
$\boldsymbol{x}^{t_0}$ = global maximizer of $g(\boldsymbol{x}, t_0)$;
for $k = 1$ to $m$ do
    $\boldsymbol{x}^{t_k}$ = Local maximizer of $g(\boldsymbol{x}; t_k)$, initialized at $\boldsymbol{x}^{t_{k-1}}$.
end
return $\boldsymbol{x}^{t_m}$.
```

4.2. Alternative Objective Function and Its Smoothing

Turning a constrained problem into an unconstrained problem can facilitate the computation of the effective gradient and Hessian of $g(\boldsymbol{x}, t)$. In this section, we consider the alternative objective function: we modify $f(\boldsymbol{x})$ by adding the penalty term $-\frac{3\tau}{4}\|\boldsymbol{x}\|_2^4$:

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

Thus we consider the following unconstrained optimization problem,

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

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.

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

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

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 12 (Smoothed Alternative Objective) *The smoothed version of the alternative objective is*

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

Its gradient and Hessian are equal to

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

and

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

Here $P_{sym}\boldsymbol{M} = \frac{\boldsymbol{M}+\boldsymbol{M}^\top}{2}$ is the projection to symmetric matrices.

The proof of this Lemma is very similar to Lemma 5 and is deferred to Appendix A.3.

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). Therefore, $g_r(\boldsymbol{x},t)$ forms a quadratic function, so it has a unique global maximizer equal to $\frac{3\tau\boldsymbol{v}+\boldsymbol{u}}{3\tau(n+2)}$, denoted as $\boldsymbol{x}^\dagger$. Notice that this vector has different norm compared to the $\boldsymbol{x}^\dagger$ in previous section.

Before we state the Theorem, we need a counterpart of the Independence Condition. We call this the Strong Independence Conjecture:

Conjecture 13 *[Strong Independence Conjecture] Suppose $\boldsymbol{T} = \tau\boldsymbol{v}^{\otimes 3} + \boldsymbol{A}$ and $\boldsymbol{u}_j = \boldsymbol{A}_{iij} + \boldsymbol{A}_{iji} + \boldsymbol{A}_{jii}$, $\delta(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{x},\boldsymbol{x},:) + \boldsymbol{A}(\boldsymbol{x},:,\boldsymbol{x}) + \boldsymbol{A}(:,\boldsymbol{x},\boldsymbol{x})$ be defined as before. With high probability, (1) $\|\boldsymbol{u}\|_2 = \Theta(n)$ and $|\langle \boldsymbol{u},\boldsymbol{v}\rangle| = O(\sqrt{n\log n})$; (2) for all $\boldsymbol{x}^{t_k}$ on the homotopy path, $\|\delta(\boldsymbol{x}^{t_k})\|_2 = \Theta(\sqrt{n})\|\boldsymbol{x}^{t_k}\|_2^2$ and $|\langle \delta(\boldsymbol{x}^{t_k}),\boldsymbol{v}\rangle| = O(\sqrt{\log n})\|\boldsymbol{x}^{t_k}\|_2^2$,*

Intuitively, this assumes that the noise is not adversarially correlated with the signal $\boldsymbol{v}$ on the entire homotopy path. The main difference between the strong independence conjecture and the weak independence conjecture is that they apply to different algorithms with different number of iterations. The strong independence conjecture applies to the general Homotopy method, which may have a large number of iterations, and thus a conjecture that depends on the number of iterations does not provide us any useful properties. We use the strong independence conjecture to analyze the general Homotopy method to gain intuitions in order to design our algorithm. The weak conjecture is for our Algorithm 1, which only has $O(\log\log n)$ rounds, and can be satisfied using the noise injection technique. Although we cannot use noise injection to prove the strong independence conjecture, similar conjectures are often used to get intuitions about optimization problems (Donoho et al., 2009; Javanmard and Montanari, 2013; Choromanska et al., 2015).

Theorem 14 *Assuming the Strong Independence Conjecture (Conjecture 13), when $\tau = n^{3/4}\log n$,*

1. *When $t \geq Cn^{-1}$ for a large enough constant C, there exists a local maximizer $\boldsymbol{x}^t$ of $g_r(\boldsymbol{x},t)$ such that $\|\boldsymbol{x}^t - \boldsymbol{x}^\dagger\|_2 = o(1)\|\boldsymbol{x}^\dagger\|_2$;*

2. *When $t < n^{-1}\log^{-2} n$, we know there are two types of local maximizers $\boldsymbol{x}^t$:*

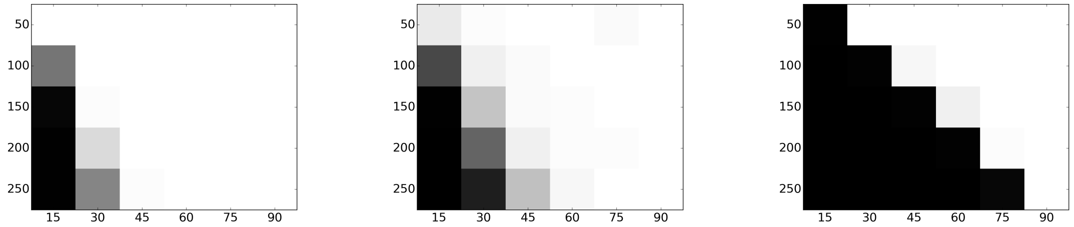
    - *$\|\boldsymbol{x}^t\|_2 = \Theta(1)$ and $\langle \boldsymbol{v},\boldsymbol{x}^t\rangle = \Theta(1)$. This corresponds to a local maximizer near the true signal $\boldsymbol{v}$.*
    - *$\|\boldsymbol{x}^t\|_2 = \Theta(n^{-\frac{1}{4}}\log^{-1} n)$ and $\langle \boldsymbol{v},\boldsymbol{x}^t\rangle = O(n^{-\frac{1}{2}}\log^{-1} n)$. These local maximizers have poor correlation with the true signal.*

3. *When $t < n^{-1}\log^{-2} n$, let $\boldsymbol{b}$ be the top eigenvector of $\nabla^2(g_r(\boldsymbol{x}^\dagger, t))$, we know* $\sin\theta(\boldsymbol{b}, \boldsymbol{v}) \leq 1/\log^2 n$.

Intuitively, this theorem shows that in the process of homotopy method, if we consider a continuous path in the sense that $t_{k+1} - t_k$ is close to 0 for all k, then (1) at the beginning, $\boldsymbol{x}^k$ is close to $\boldsymbol{x}^\dagger$; (2) at some point k^*, $\boldsymbol{x}^{k^*}$ is a saddle point in the function $g(\boldsymbol{x}, t_{k^*+1})$ and 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 phenomenon is illustrated in Figure 1:

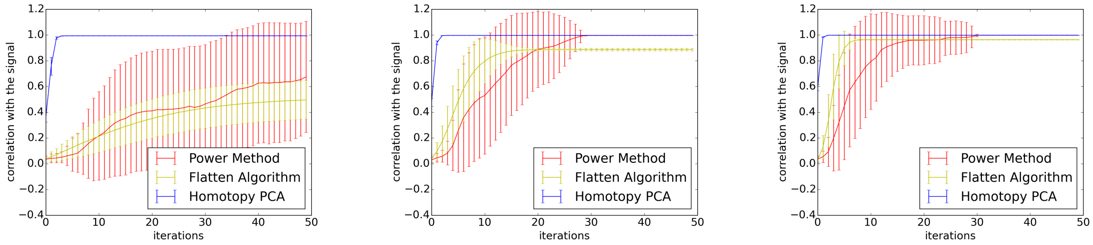
Figure 1(a) has large smoothing parameter, and the function has a unique local/global maximizer. Figure 1(b) has medium smoothing parameter, the original global maximizer now behaves like a local minimizer in one dimension, but it in general could be a saddle point in high dimensions. The Hessian at 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.

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 Richard and Montanari (2014), where they show it works when $\tau = \Omega(n)$. Hopkins et al. (2015) 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 Hopkins et al. (2015), 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

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.

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.

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.

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

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

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

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

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.

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.

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

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

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

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.

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.

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

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

Michel Ledoux. Deviation inequalities on largest eigenvalues. In *Geometric aspects of functional analysis*, pages 167–219. Springer, 2007.

Hossein Mobahi. Closed form for some gaussian convolutions. *arXiv:1602.05610*, 2016.

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

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, 2015b.

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.

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.

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

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.

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

Ryota Tomioka and Taiji Suzuki. Spectral norm of random tensors. *arXiv preprint arXiv:1407.1870*, 2014.

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.

Appendix A. Omitted Proofs

A.1. Omitted Proof in Section 2

Lemma 15 (Lemma 5 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, let $\boldsymbol{z}$ be a vector where $\boldsymbol{z}_j = \sum_{i=1}^d (\boldsymbol{T}_{iij} + \boldsymbol{T}_{iji} + \boldsymbol{T}_{jii})$, then we have $\boldsymbol{z} = 3\tau\boldsymbol{v} + \boldsymbol{u}$.

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 the linear term on $\boldsymbol{x}$, and it 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

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

■

A.2. Omitted Proof in Section 3

Theorem 16 (Theorem 7 restated) *When $\tau \geq Cn^{3/4}\log n$ for a large enough constant C, under the Independence Condition (Condition 6), Algorithm 1 finds a vector $\boldsymbol{x}^m$ such that $\langle \boldsymbol{x}^m, \boldsymbol{v}\rangle \geq 0.8$ in $O(\log\log n)$ iterations.*

Proof We first show the initial maximizer $\boldsymbol{x}^0$ already has a nontrivial correlation with $\boldsymbol{v}$. Recall $\boldsymbol{x}^0 = \frac{3\tau\boldsymbol{v}+\boldsymbol{u}}{\|3\tau\boldsymbol{v}+\boldsymbol{u}\|_2}$. Note that if τ is very large such that $\|3\tau\boldsymbol{v}\|_2 \geq 10\|u\|_2$, then we already have

Algorithm 4: Tensor PCA by Homotopy Initialization

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

$\langle \boldsymbol{x}^0, \boldsymbol{v}\rangle \geq 0.8$. Later we will show that whenever $\langle \boldsymbol{x}^i, \boldsymbol{v}\rangle \geq 0.8$ all later iterations have the same property.

Therefore, we are left with the case when $\|\boldsymbol{u}\|_2 \geq 0.1\|3\tau\boldsymbol{v}\|_2$ (this implies $\tau \leq O(n)$). In this case, by Condition 6 we know $|\langle \boldsymbol{u}, \boldsymbol{v}\rangle| = O(\sqrt{nm\log n})$ and $\|\boldsymbol{u}\|_2 = O(n\sqrt{m})$, therefore

$$\|3\tau\boldsymbol{v} + \boldsymbol{u}\|_2 \in \left[\sqrt{\|\boldsymbol{u}\|_2^2 + \|3\tau\boldsymbol{v}\|_2^2 - O(\tau\sqrt{nm\log n})}, \sqrt{\|\boldsymbol{u}\|_2^2 + \|3\tau\boldsymbol{v}\|_2^2 + O(\tau\sqrt{nm\log n})}\right]$$

Therefore, $\|3\tau\boldsymbol{v} + \boldsymbol{u}\|_2 = \Theta(n\sqrt{m})$. Assume $\tau \geq Cn^{3/4}\log^c n$ for large enough C (where we will later show $c = 1$ suffices)

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

Now let us consider the first step of power method. Let $\hat{\boldsymbol{x}}^1$ be the vector before normalization. Observe that $\hat{\boldsymbol{x}}^1 = 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0\rangle^2\boldsymbol{v} + \delta(\boldsymbol{x}^0)$. By Condition 6 we have bounds on $\|\delta(\boldsymbol{x}^0)\|$ and $|\langle\delta(\boldsymbol{x}^0), v\rangle|$, therefore we have

$$\langle \hat{\boldsymbol{x}}^1, \boldsymbol{v}\rangle = 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0\rangle^2 + \langle\delta(\boldsymbol{x}^0), \boldsymbol{v}\rangle \in \left[3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0\rangle^2 - O(\sqrt{m\log n}), 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0\rangle^2 + O(\sqrt{m\log n})\right].$$

Note that when $\tau \geq Cn^{3/4}\log^c n$ and $\log^c n \gg m$, the first term is much larger than $\sqrt{m\log n}$. Hence for the first iteration, we have $\langle \hat{\boldsymbol{x}}^1, \boldsymbol{v}\rangle \geq (3 - o(1))\tau\langle \boldsymbol{v}, \boldsymbol{x}^0\rangle^2 \geq 2Cn^{\frac{1}{4}} \cdot \log^{3c} n/m$.

Similar as before, when $\|\delta(\boldsymbol{x}^0)\|_2 \leq 0.1\|3\tau\langle v, \boldsymbol{x}^0\rangle^2 v\|_2$, we already have $\langle \boldsymbol{x}^1, \boldsymbol{v}\rangle \geq 0.8$. On the other hand, if $\|\delta(\boldsymbol{x}^0)\|_2 \geq 0.1\|3\tau\langle v, \boldsymbol{x}^0\rangle^2 v\|_2$, in this case, by Condition 6 we know $\|\delta(\hat{\boldsymbol{x}}^0)\| = O(\sqrt{nm})$. We again have $\|\hat{\boldsymbol{x}}^1\|_2 \in \sqrt{\|\delta(\boldsymbol{x}^0)\|_2^2 + \|3\tau\langle \boldsymbol{v}, \boldsymbol{x}^0\rangle^2\boldsymbol{v}\|_2^2 \pm O(\tau\langle \boldsymbol{v}, \boldsymbol{x}^0\rangle^2\sqrt{nm})}$. Therefore, $\|\hat{\boldsymbol{x}}^1\|_2 = O(\sqrt{nm})$. Combining the bounds for the norm of $\hat{\boldsymbol{x}}^1$ and its correlation with $\boldsymbol{v}$,

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

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 $\langle \boldsymbol{x}^p, \boldsymbol{v}\rangle \geq 0.8$ or

$$\langle \boldsymbol{x}^p, \boldsymbol{v}\rangle \geq n^{-\frac{1}{4}} \log^{3^p c} n / m^{2^p - \frac{1}{2}}.$$

Initially, for $p = 0$, we have already proved the induction hypothesis.

Now assume the induction hypothesis is true for p. In the next iteration, let $\hat{\boldsymbol{x}}^{p+1}$ be the vector before normalization. Similar as before we have $\hat{\boldsymbol{x}}^{p+1} = 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2 \boldsymbol{v} + \delta(\boldsymbol{x}^p)$.

When $\langle \boldsymbol{x}^p, \boldsymbol{v}\rangle \geq 0.8$, by Condition 6 we know the norm of $\|\delta(\boldsymbol{x}^p)\|$ is much smaller than $3\tau\langle \boldsymbol{x}^p, \boldsymbol{v}\rangle^2$. Therefore we still have $\langle \boldsymbol{x}^{p+1}, \boldsymbol{v}\rangle \geq 0.8$.

In the other case, we follow the same strategy as the first step. By Condition 6 we can compute the correlation between $\hat{\boldsymbol{x}}^{p+1}$ and $\boldsymbol{v}$:

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

For the norm of $\hat{\boldsymbol{x}}^{p+1}$, notice that the first term $3\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2 \boldsymbol{v}$ 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 Condition, therefore

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

If $3\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2 \geq \Delta\sqrt{nm}$, then $\|\hat{\boldsymbol{x}}^{p+1}\|_2 \leq (3 + \Delta')\tau\langle \boldsymbol{v}, \boldsymbol{x}^p\rangle^2$, where Δ' is a constant that is smaller than 0.1 when Δ is large enough. Therefore in this case $\langle \frac{\hat{\boldsymbol{x}}^{p+1}}{\|\hat{\boldsymbol{x}}^{p+1}\|_2}, \boldsymbol{v}\rangle \geq 0.8$. Thus we successfully recover $\boldsymbol{v}$ in the next step.

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

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

If we select $c = 1$, after $m = O(\log \log n)$ rounds, we have $\langle \boldsymbol{x}^p, \boldsymbol{v}\rangle \geq n^{-\frac{1}{4}} \log^{3^p c} n / m^{2^p - \frac{1}{2}} \geq 0.8$, therefore we must always be in the first case. As a result $\langle \boldsymbol{x}^m, \boldsymbol{v}\rangle \geq 0.8$. ■

Lemma 17 (Lemma 8 restated) *Let the sequence $\boldsymbol{T}^0, \cdots, \boldsymbol{T}^{m-1}$ be generated according to Section 3.1. Let $\boldsymbol{Q}^i = \tau \boldsymbol{v}^{\otimes 3} + \boldsymbol{C}^i$, where $\boldsymbol{C}^i$'s are tensors with independent Gaussian entries. Each entry in $\boldsymbol{C}^i$ is distributed as $N(0, m)$. The two sets of variables $\{\boldsymbol{T}^i\}$ and $\{\boldsymbol{Q}^i\}$ has the same distribution.*

Proof Note that both distributions are multivariate Gaussians. Therefore we only need to show that they have the same first and second moments.

For the first moment, this is easy, we have $\mathbb{E}[\boldsymbol{T}^p] = \tau \cdot v^{\otimes 3}$ and $\mathbb{E}[\boldsymbol{Q}^p] = \tau \cdot v^{\otimes 3}$ for all p.

For the second moment (covariance), we consider the covariance between T^p_{ijk} and $T^q_{i'j'k'}$. Note that for the distribution Q, as long as the 4 tuple $(p, i, j, k) \neq (q, i', j', k')$ the correlation is 0. We first show when $(i,j,k) \neq (i',j',k')$ we have

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

Hence for these variables the two distributions have the same covariance.

Next we consider the case $p \neq q$,

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

The covariance for these entries also match.

Finally we need to consider the variance for each entry of $\boldsymbol{T}^p$ and $\boldsymbol{Q}^p$. To do that we compute the Variance of $\boldsymbol{T}^p_{ijk}$

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

This is also the same as the variance of Q^p_{ijk}. Therefore the two multivariate Gaussians have the same mean and covariance, and must be the same distribution. ■

Lemma 18 (Lemma 9 restated) *Let $\boldsymbol{T}^p$ be generated according to Algorithm 2 and $\boldsymbol{A}^p = \boldsymbol{T}^p - \tau\boldsymbol{v}^{\otimes 3}$. Let $\boldsymbol{u}^0$ be a vector such that $\boldsymbol{u}^0_j = \sum_i \boldsymbol{A}^0_{iij} + \boldsymbol{A}^0_{iji} + \boldsymbol{A}^0_{jii}$, and $\delta^p(\boldsymbol{x}^p) = \boldsymbol{A}^p(\boldsymbol{x}^p, \boldsymbol{x}^p, :) + \boldsymbol{A}^p(\boldsymbol{x}^p, :, \boldsymbol{x}^p) + \boldsymbol{A}^p(:, \boldsymbol{x}^p, \boldsymbol{x}^p)$. With high probability, (1) $\|\boldsymbol{u}^0\|_2 = \Theta(n\sqrt{m})$ and $|\langle \boldsymbol{u}^0, \boldsymbol{v}\rangle| = O(\sqrt{nm\log n})$; (2) for the sequence computed by Algorithm 2, $\boldsymbol{x}^0, \boldsymbol{x}^1, \cdots, \boldsymbol{x}^{m-1}$, $\forall\, 0 \leq p \leq m-1$, $\|\delta^p(\boldsymbol{x}^p)\|_2 = \Theta(\sqrt{nm})\|\boldsymbol{x}^p\|_2^2$ and $|\langle \delta^p(\boldsymbol{x}^p), \boldsymbol{v}\rangle| = O(\sqrt{m\log n})\|\boldsymbol{x}^p\|_2^2$. As a result Condition 6 is satisfied.*

Proof

Since by Lemma 8, we know the noise tensors $\boldsymbol{A}^p$ used in p-th step behave exactly the same as independent Gaussian tensors. The vectors $\boldsymbol{u}^0$ and $\delta(x^p)$ are therefore spherical Gaussian random variables conditioned on any value of $\boldsymbol{x}^i$. Therefore we can prove this lemma by standard Gaussian concentration results.

Claim 19 *(Laurent and Massart, 2000) Suppose $\boldsymbol{x}$ is a d-dimensional spherical Gaussian, then*

$$\Pr[|\|\boldsymbol{x}\|^2 - \mathbb{E}[\|\boldsymbol{x}\|^2]| \geq \frac{1}{2}\mathbb{E}[\|\boldsymbol{x}\|^2]] \leq e^{-\Omega(d)}.$$

Also, for any fixed vector $\mathbf{v}$, $\langle \boldsymbol{x}, \boldsymbol{v}\rangle$ is also a Gaussian distribution that satisfies

$$\Pr[|\langle \boldsymbol{x}, \boldsymbol{v}\rangle| \geq t\sqrt{\mathbb{E}[\langle \boldsymbol{x}, \boldsymbol{v}\rangle^2]}] \leq e^{-\Omega(t^2)}.$$

For terms like $\|\boldsymbol{u}^p\|$ 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 $\langle \boldsymbol{u}^p, \boldsymbol{v}\rangle$ and $\langle \delta(\boldsymbol{x}^p), \boldsymbol{v}\rangle$, we know they are just Gaussian distributions and is always bounded by $O(\sigma\sqrt{\log n})$ with high probability. Therefore we only need to compute the expected norms of these vectors.

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

Therefore by Claim 19 we have $\|u\|_2 = \Theta(n\sqrt{m})$ with high probability.

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

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

This means $\langle u, v\rangle$ is a Gaussian random variable with variance $\sigma^2 = \Theta(nm)$, therefore for any constant C', with probability at least $1 - n^{-C'}$ we know $|\langle u, v\rangle| \leq O(\sqrt{nm\log n})$. We can apply union bound over all p and get the desired result.

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

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

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

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

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 20 (Lemma 12 restated) *The smoothed version of the alternative objective is*

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

Its gradient and Hessian are equal to

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

and

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

Proof Similar to Lemma 5, 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:

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{x} + t\boldsymbol{y}\|_2^4] &= \mathbb{E}[(\|\boldsymbol{x}\|_2^2 + 2t\langle \boldsymbol{x}, \boldsymbol{y}\rangle + t^2\|\boldsymbol{y}\|_2^2)^2] \\ &= \mathbb{E}[\|\boldsymbol{x}\|_2^4 + 4t^2\langle \boldsymbol{x}, \boldsymbol{y}\rangle^2 + t^4\|\boldsymbol{y}\|_2^4 + 2t^2\|x\|^2\|y\|^2] \\ &= \|x\|_2^4 + t^2(2n+4)\|x\|_2^2 + t^4(n^2+2n) = \|x\|_2^4 + 2t^2(n+2)\|x\|_2^2 + t^4(n^2+2n).\end{aligned}$$

Here in the second equation we omitted all the odd order terms for $\boldsymbol{y}$ because those terms have expectation 0. The final step uses the moments of Gaussians.

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. ■

Before trying to characterize the local maxima on the homotopy path, let us first prove the following property for the matrix $P_{sym}[\boldsymbol{A}(\boldsymbol{x}, :, :) + \boldsymbol{A}(:, \boldsymbol{x}, :) + \boldsymbol{A}(:, :, \boldsymbol{x})]$.

Lemma 21 *Let* $H(\boldsymbol{x}) = P_{sym}[\boldsymbol{A}(\boldsymbol{x},:,:) + \boldsymbol{A}(:,\boldsymbol{x},:) + \boldsymbol{A}(:,:,\boldsymbol{x})]$*, there exists constants* c^-, c^+ *such that with probability at least* $1 - \exp(-\Omega(n))$*, for any unit vector* $\boldsymbol{x}$ *we have*

$$c^-\sqrt{n} \le \lambda_{max} \le c^+\sqrt{n}.$$

Proof For the upperbound, we use the bound on tensor spectral norm. Tomioka and Suzuki (2014) proved that for a random Gaussian tensor $\boldsymbol{A}$, with probability at least $1 - \exp(-\Omega(n))$ we know for any vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$, $|\boldsymbol{A}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})| \le O(\sqrt{n})$. Therefore for any unit vector $\boldsymbol{y}$, $|\boldsymbol{y}^\top H(\boldsymbol{x})\boldsymbol{y}| = |\boldsymbol{A}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{y}) + \boldsymbol{A}(\boldsymbol{y},\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{A}(\boldsymbol{y},\boldsymbol{y},\boldsymbol{x})| \le O(\sqrt{n})$.

For the lowerbound, we use the distribution of the largest eigenvalue of Gaussian Orthogonal Ensemble. Suppose $\boldsymbol{M}$ is a random matrix whose entries are i.i.d. standard Gaussians, then the symmetric matrix $\frac{\boldsymbol{M}+\boldsymbol{M}^\top}{\sqrt{2}}$ is distributed according to the Gaussian Orthogonal Ensemble. Let $P_{\boldsymbol{x}^\perp}$ be the projection operator to the orthogonal subspace of $\boldsymbol{x}$, then the key observation is $P_{\boldsymbol{x}^\perp}H(\boldsymbol{x})P_{\boldsymbol{x}^\perp}$ is (up to a constant scaling) distributed as a Gaussian Orthogonal Ensemble of dimension $(n-1)\times(n-1)$. To see this, the easiest way is to observe that Gaussians are invariant under rotation, so we can take $\boldsymbol{x} = \boldsymbol{e}_1$. Now for $i,j = \{2,3,...,n\}$, $[P_{\boldsymbol{x}^\perp}H(\boldsymbol{x})P_{\boldsymbol{x}^\perp}]_{i,j} = \boldsymbol{A}_{1ij} + \boldsymbol{A}_{1ji} + \boldsymbol{A}_{i1j} + \boldsymbol{A}_{j1i} + \boldsymbol{A}_{ij1} + \boldsymbol{A}_{ji1}$. The random entries $1ij, i1j, ij1$ do not overlap because $i,j \ne 1$. Therefore the matrix is the sum of three Gaussian Orthogonal Ensembles, and by property of Gaussians that is equivalent to $\sqrt{3}$ times a Gaussian Orthogonal Ensemble. Now, using the result in Ledoux (2007), we know for any fixed $\boldsymbol{x}$, $\Pr[\lambda_{max}(P_{\boldsymbol{x}^\perp}H(\boldsymbol{x})P_{\boldsymbol{x}^\perp}) \le \sqrt{n}/2] \le 1 - \exp(-\Omega(n^2))$. By standard covering argument (the ϵ-net for n dimensional vectors have size $(n/\epsilon)^{O(n)}$ which is much smaller than $\exp(-\Omega(n^2))$), we know with high probability for all $\boldsymbol{x}$ $\lambda_{max}(P_{\boldsymbol{x}^\perp}H(\boldsymbol{x})P_{\boldsymbol{x}^\perp}) \ge \sqrt{n}/2$. The lemma follows immediately because $\lambda_{max}(H(\boldsymbol{x})) \ge \lambda_{max}(P_{\boldsymbol{x}^\perp}H(\boldsymbol{x})P_{\boldsymbol{x}^\perp})$. ■

Now we are ready to prove Theorem 14. To capture the properties of the homotopy path, we break it into three lemmas.

Lemma 22 *When* $\tau = n^{3/4}\log n$*,* $t \ge Cn^{-1}$ *for large enough constant* C*, 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 Recall according to the objective we chose, the maximizer at infinity $\boldsymbol{x}^\dagger$ can be computed explicitly and we know $\boldsymbol{x}^\dagger = \frac{3\tau\boldsymbol{v}+\boldsymbol{u}}{3\tau(n+2)}$. By Conjecture 13, we can estimate the norm and correlation with $\boldsymbol{v}$:

$$\|\boldsymbol{x}^\dagger\|_2 = \Theta(n^{-3/4}\log^{-1} n), \quad \langle \boldsymbol{x}^\dagger, \boldsymbol{v}\rangle = (1 \pm o(1))/n.$$

We shall first prove in the region $\mathcal{B} = \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^\dagger\|_2 \le \frac{1}{2}\|\boldsymbol{x}^\dagger\|_2, \langle \boldsymbol{x}, \boldsymbol{v}\rangle \le 10/n\}$, the Hessian of the objective function is always negative definite. By standard analysis in convex optimization, this in particular implies two things: 1. There can be at most one local maximizer in this region; 2. If the function is μ-strongly-concave ($\nabla^2 g(\boldsymbol{x},t) \succeq -\mu I$), and a point $\boldsymbol{x}$ has $\|\nabla g(\boldsymbol{x},t)\| \le \epsilon$, then there is a local maximizer within ϵ/μ. This particular implies if there is a point $\boldsymbol{x}$ in the interior $\mathcal{B}' = \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^\dagger\|_2 \le \frac{1}{4}\|\boldsymbol{x}^\dagger\|_2, \langle \boldsymbol{x}, \boldsymbol{v}\rangle \le 2/n\}$ such that $\nabla g(\boldsymbol{x},t)$ is very small, then there must exist a local maximizer in $\mathcal{B}$.

By Lemma 12, we know the Hessian is equal to:

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

In the region we are interested in, since $\langle \boldsymbol{v}, \boldsymbol{x}\rangle \le 10/n \le t^2(n+2)/2$ when C is large enough, we have the first term

$$-3\tau((\|\boldsymbol{x}\|_2^2 + t^2(n+2))\boldsymbol{I} - 2\langle \boldsymbol{v}, \boldsymbol{x}\rangle \boldsymbol{v}\boldsymbol{v}^T + 2\boldsymbol{x}\boldsymbol{x}^T) \preceq -1.5\tau t^2(n+2)\boldsymbol{I}.$$

On the other hand, for the second part we know by Lemma 21

$$P_{sym}[\boldsymbol{A}(\boldsymbol{x},:,:) + \boldsymbol{A}(:,\boldsymbol{x},:) + \boldsymbol{A}(:,:,\boldsymbol{x})] \preceq \frac{1}{2}c^+\sqrt{n}\|\boldsymbol{x}^\dagger\|_2 I.$$

By our choice of parameters, $\tau t^2(n+2) = \Omega(n^{-1/4}\log n)$, and $\sqrt{n}\|\boldsymbol{x}^\dagger\|_2 = \Theta(n^{-1/4}\log^{-1} n)$, therefore the first term dominates and we know the Hessian $\nabla^2 g_r(\boldsymbol{x}, t) \preceq -\tau t^2(n+2)\boldsymbol{I}$.

When t is a large polynomial of n (e.g. $t = n^{10}$), simple calculation shows the optima $\boldsymbol{x}^t$ is very close to $\boldsymbol{x}^\dagger$, and we have $\boldsymbol{x}^t \in \mathcal{B}'$. When $C/n \le t < n^{10}$, let $t_0 = n^{10}$, select $t_1, t_2, ..., t_q$ such that $t_q = t$, and t_i, t_{i+1} are close enough that if $\boldsymbol{x}^{t_i} \in \mathcal{B}'$, by strong concavity we can get $\boldsymbol{x}^{t_{i+1}}$ exists and $\boldsymbol{x}^{t_{i+1}} \in \mathcal{B}$. We will prove $\boldsymbol{x}^{t_i} \in \mathcal{B}'$ by induction. The base case is already done.

Suppose $\boldsymbol{x}^{t_{i-1}} \in \mathcal{B}'$, we know that $\boldsymbol{x}^{t_i} \in \mathcal{B}$. We will use the first order condition to refine our knowledge about $\boldsymbol{x}^{t_i}$ and show $\boldsymbol{x}^{t_i} \in \mathcal{B}'$. From (5), we can derive the expression of stationary points,

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

Note that $\boldsymbol{x}^{t_i}$ is a stationary point on homotopy path, so it should satisfy Conjecture 13. We also know it is in $\mathcal{B}$.

Since $t \ge Cn^{-1}$, $\|\tau\langle \boldsymbol{v}, \boldsymbol{x}^{t_i}\rangle^2\boldsymbol{v}\|_2 = \Theta(n^{-\frac{5}{4}}\log n)$, $\|t^2(3\tau\boldsymbol{v}+\boldsymbol{u})\|_2 \ge \Omega(n^{-1})$ and $\|\delta(\boldsymbol{x}^{t_i})\|_2 = \Theta(n^{-1}\log^{-2} n)$. Therefore, if we let $\boldsymbol{w} = 3\tau\langle \boldsymbol{v}, \boldsymbol{x}^{t_i}\rangle^2\boldsymbol{v} + \delta(\boldsymbol{x}^{t_i})$ we know $\|\boldsymbol{w}\|_2 \le o(1)\|t^2(3\tau\boldsymbol{v} + \boldsymbol{u})\|_2$. The middle term dominates the numerator. Moreover, $t^2(n+2) \ge \Omega(n^{-1})$ and $\|\boldsymbol{x}^{t_i}\|_2^2 = \Theta(n^{-3/2}\log^{-2} n)$, and thus, $t^2 n$ dominates the denominator. Now we have

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

Since $\epsilon = o(1)$ and $\|w\|_2 \le o(1)\|t^2(3\tau\boldsymbol{v} + \boldsymbol{u})\|_2$, we know the two additional term has norm $o(1)\|\boldsymbol{x}^\dagger\|_2$, therefore $\boldsymbol{x}^{t_i}$ is very close to $\boldsymbol{x}^\dagger$.

Next we bound the correlation with $\boldsymbol{v}$. We know $\langle \boldsymbol{v}, \boldsymbol{x}^{t_i}\rangle \le 10/n$ because $\boldsymbol{x}^{t_i} \in \mathcal{B}$. Also, the correlation between $|\langle \boldsymbol{u}, \boldsymbol{v}\rangle| = O(\sqrt{n\log n})$ and $|\langle \delta(\boldsymbol{x}^{t_i}), \boldsymbol{v}\rangle| = O(n^{-3/2}\log^{-3/2} n)$ are negligible due to Conjecture 13, therefore we have

$$\langle \boldsymbol{x}^{t_i}, \boldsymbol{v}\rangle \le \frac{3\tau(10/n)^2 + 3\tau t^2}{3\tau t^2(n+2)(1+\epsilon)} \approx \frac{10^2 + C^2}{C^2 n} \le 2/n.$$

Here the inequality holds as long as C is large enough. Therefore $x^{t_i} \in \mathcal{B}'$ and we finish the induction.

■

Next lemma shows what happens after the phase transition, when t is small.

Lemma 23 *When $\tau = n^{3/4}\log n$, $t = n^{-1}\varepsilon(n)$, 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 $\langle \boldsymbol{v}, \boldsymbol{x}^t\rangle \le O(n^{-\frac{1}{2}}\log^{-2} n)$;*

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 $\boldsymbol{x}^t$:

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

Note that $\boldsymbol{x}^t$ should satisfy the conditions in Conjecture 13, in particular $|\langle \delta(\boldsymbol{x}^t, \boldsymbol{v}\rangle| \le O(1)\|\boldsymbol{x}^t\|_2^2$. Also, by Conjecture 13 we know $t^2\langle \boldsymbol{u}, \boldsymbol{v}\rangle = t^2 O(\sqrt{n\log n}) \ll 3\tau t^2$, so it is negligible in scale analysis. Therefore,

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

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

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

where the cross term $\eta(\boldsymbol{x}^t)$

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

is negligible compared to the other terms. We again have the bound on $\|\delta(\boldsymbol{x}^t)\|_2$ from Conjecture 13 and therefore

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

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

Case 1: $\|\boldsymbol{x}^t\|_2^2 \ge t^2 n$:

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 $\lambda_{max}(P_{sym}(\boldsymbol{A}(\boldsymbol{x}^t, :, :) + \boldsymbol{A}(:, \boldsymbol{x}^t, :) + \boldsymbol{A}(:, :, \boldsymbol{x}^t)))$. By Lemma 21 we have $\tau\|\boldsymbol{x}^t\|_2^2 > \Theta(\sqrt{n})\|\boldsymbol{x}^t\|_2$, which implies $\|\boldsymbol{x}^t\|_2 = \Omega(n^{-\frac{1}{4}}\log^{-1} n)$. As a result, $\Theta(n)\|\boldsymbol{x}^t\|_2^4$ dominates $t^4\Theta(n^2)$ in the nominator of (9). Henceforth, we have

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

We know $\|\boldsymbol{x}^t\|_2^2$ must be within constant factor to either $\frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^4}{\|\boldsymbol{x}^t\|_2^4}$ or $n^{-\frac{1}{2}}\log^{-1} n$. These two cases are discussed below

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

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

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

(2) If $\|\boldsymbol{x}^t\|_2^3 = \Theta(1)\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2$, plug it into (8):

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

Thus, we can conclude both $\|\boldsymbol{x}^t\|_2$ and $\langle \boldsymbol{v},\boldsymbol{x}^t\rangle$ are bounded by absolute constants.

Case 2: $t^2 n \geq \|\boldsymbol{x}^t\|_2^2$

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 $\lambda_{max}(P_{sym}(\boldsymbol{A}(\boldsymbol{x}^t,:,:)+\boldsymbol{A}(:,\boldsymbol{x}^t,:)+\boldsymbol{A}(:,:,\boldsymbol{x}^t)))$. By Lemma 21, we have $\tau t^2 n > \Theta(\sqrt{n})\|\boldsymbol{x}^t\|_2$, which implies $\|\boldsymbol{x}^t\|_2 = t^2\tau O(n^{1/2})$. As a result, $3\tau t^2$ dominates $O(\sqrt{\log n})\|\boldsymbol{x}^t\|_2^2$ in the nominator of (8). Henceforth, we have

$$\langle \boldsymbol{v},\boldsymbol{x}^t\rangle = C_1\frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2}{t^2 n} + C_2\frac{1}{n},$$

where both C_1, C_2 are constants within $1 \pm 2/3$. Notice that if $\frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2}{t^2 n} \geq n^{-1}$, then $\langle \boldsymbol{v},\boldsymbol{x}^t\rangle = \Theta(t^2 n)$, implying $\frac{\langle \boldsymbol{v},\boldsymbol{x}^t\rangle^2}{t^2 n} = \Theta(t^2 n) = \Theta(\frac{\varepsilon^2(n)}{n}) \ll n^{-1}$. This is a contradiction, so we know, $\langle \boldsymbol{v},\boldsymbol{x}^t\rangle$ can only be $\Theta(n^{-1})$.

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

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

This contradicts with $\|\boldsymbol{x}^t\|_2 = t^2\tau O(\sqrt{n}) = O(n^{-\frac{3}{4}}\log^{-3} n)$. There cannot be a local maximizer in this case. ■

Finally we show that the Hessian is correlated with the correct vector $\boldsymbol{v}$ near the threshold.

Lemma 24 *For $t = n^{-1}\varepsilon(n)$, where $\varepsilon(n) = O(\log^{-2} n)$, let **b** be the top eigenvector of $\nabla^2(g_r(\boldsymbol{x}^\dagger, t))$, we know $\sin\theta(\boldsymbol{b},\boldsymbol{v}) \leq 1/\log^2 n$.*

Proof Recall the formula for the Hessian (6),

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

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 $\|\boldsymbol{x}^\dagger\|_2^2 + t^2 n = O(n^{-1}\log^{-4} n)$. By Lemma 21 the spectral norm of $P_{sym}(\boldsymbol{A}(\boldsymbol{x}^\dagger,:,:)+\boldsymbol{A}(:,\boldsymbol{x}^\dagger,:)+\boldsymbol{A}(:,:,\boldsymbol{x}^\dagger))$ is $\Theta(n^{-\frac{1}{4}}\log^{-1} n)$. Thus, we can write the Hessian as

$$\nabla^2 g_r(\boldsymbol{x}^\dagger,t) = 6\tau\langle \boldsymbol{v},\boldsymbol{x}^\dagger\rangle\boldsymbol{v}\boldsymbol{v}^\top + \boldsymbol{E},$$

where the main term $\boldsymbol{v}\boldsymbol{v}^T$ has coefficient $6\tau\langle \boldsymbol{v}, \boldsymbol{x}^\dagger \rangle = \Theta(n^{-\frac{1}{4}} \log n)$, and the spectral norm of E is bounded by $O(n^{-1/4} \log^{-1} n)$. By Davis Kahan theorem we know $\sin \theta(\boldsymbol{b}, \boldsymbol{v}) \leq O(1/\log^2 n)$, that is, the top eigenvector of the Hessian is $O(1/\log^2 n)$ close to $\boldsymbol{v}$. ■