JMLR: Workshop and Conference Proceedings vol 49:1–22, 2016

Efficient approaches for escaping higher order saddle points in non-convex optimization

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

Rong Ge RONGGE@CS.DUKE.EDU
Duke University

Abstract

Local search heuristics for non-convex optimizations are popular in applied machine learning. However, in general it is hard to guarantee that such algorithms even converge to a *local minimum*, due to the existence of complicated saddle point structures in high dimensions. Many functions have *degenerate* saddle points such that the first and second order derivatives cannot distinguish them with local optima. In this paper we use higher order derivatives to escape these saddle points: we design the first efficient algorithm guaranteed to converge to a third order local optimum (while existing techniques are at most second order). We also show that it is NP-hard to extend this further to finding fourth order local optima.

Keywords: Non-convex optimization, degenerate saddle points, higher order conditions for local optimality, trust region methods.

1. Introduction

Recent trend in applied machine learning has been dominated by the use of large-scale non-convex optimization, e.g. deep learning. However, analyzing non-convex optimization in high dimensions is very challenging. Current theoretical results are mostly negative regarding the hardness of reaching the globally optimal solution.

Less attention is paid to the issue of reaching a locally optimal solution. In fact, even this is computationally hard in the worst case (Nie, 2015). The hardness arises due to diversity and ubiquity of critical points in high dimensions. In addition to local optima, the set of critical points also consists of saddle points, which possess directions along which the objective value improves. Since the objective function can be arbitrarily bad at these points, it is important to develop strategies to escape them, in order to reach a local optimum.

The problem of saddle points is compounded in high dimensions. Due to curse of dimensionality, the number of saddle points grows exponentially for many problems of interest, e.g. (Auer et al., 1996; Cartwright and Sturmfels, 2013; Auffinger et al., 2013). Ordinary gradient descent can be stuck in a saddle point for an arbitrarily long time before making progress. A few recent works have addressed this issue, either by incorporating second order Hessian information (Nesterov and Polyak, 2006) or through noisy stochastic gradient descent (Ge et al., 2015). These works however require the Hessian matrix at the saddle point to have a strictly negative eigenvalue, termed as the *strict saddle* condition. The time to escape the saddle point depends (polynomially) on this negative eigenvalue. Some

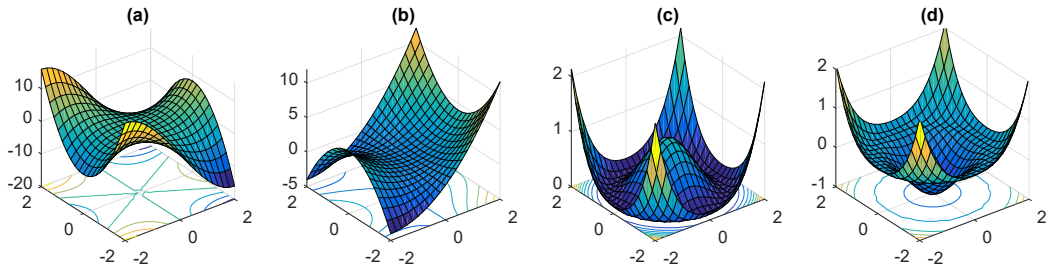

Figure 1: Examples of Degenerate Saddle Points: (a) Monkey Saddle $-3x^2y+y^3$, $(0,0)$ is a second order local minimum but not third order local minimum; (b) x^2y+y^2, $(0,0)$ is a third order local minimum but not fourth order local minimum; (c) "wine bottle", the bottom of the bottle is a connected set with degenerate Hessian; (d) "inverted wine bottle": the points on the circle with degenerate Hessian are actually saddle points and not local minima.

structured problems such as complete dictionary learning, phase retrieval and orthogonal tensor decomposition possess this property (Sun et al., 2015).

On the other hand, for problems without the strict saddle property, the above techniques can converge to a saddle point, which is disguised as a local minimum when only first and second order information is used. We address this problem in this work, and extend the notion of second order optimality to higher order optimality conditions. We propose a new efficient algorithm that is guaranteed to converge to a third order local minimum, and show that it is NP-hard to find a fourth order local minimum.

Our results are relevant for a wide range of non-convex problems which possess degenerate critical points. At these points, the Hessian matrix is singular. Such points arise due to symmetries in the optimization problem, e.g., permutation symmetry in a multi-layer neural network. Singularities also arise in over-specified models, where the model capacity (such as the number of neurons in neural networks) exceeds the complexity of the target function. Here, certain neurons can be eliminated (i.e. have weights set to zero), and such critical points possess the so-called *elimination singularity* (Wei et al., 2008). Alternatively, two neurons can have the same weight, and this is known as *overlap singularity* (Wei et al., 2008). The Hessian matrix is singular at such critical points. This behavior is limited not just to neural networks, but has also been studied in overspecified Gaussian mixtures, radial basis function networks, ARMA models of time series (Amari et al., 2006; Wei et al., 2008), and student-teacher networks, also known as *soft committee models* (Saad and Solla, 1995; Inoue et al., 2003).

The current trend in practice is to incorporate overspecified models (Giles, 2001). Theoretically, bad local optima are guaranteed to disappear in neural networks under massive levels of overspecification (Safran and Shamir, 2015). On the other hand, as discussed above, the saddle point problem is compounded in these overspecified models. Empirically, the presence of singular saddle points is found to slow down learning substantially (Saad and Solla, 1995; Inoue et al., 2003; Amari et al., 2006; Wei et al., 2008). Intuitively, these

singular saddle points are surrounded by *plateaus* or flat regions with a sub-optimal objective value. For these regions neither the gradient or Hessian information can lead to a direction that improves the function value. Therefore they can "fool" the (ordinary) first and second order algorithms and they may stuck there for long periods of time. Higher order derivatives are needed to classify the point as either a local optimum or a saddle point. In this work, we tackle this challenging problem of escaping such higher order saddle points.

1.1. Summary of Results

We call a point x a p^{th} order local minimum if for any nearby point y $f(x)-f(y) \leq o(\|x-y\|^p)$ (see Definition 3).

We give a necessary and sufficient condition for a point x to be a third order local minimum (see Section 4). Similar conditions (for even higher order) have been discussed in previous works, however their algorithmic implications were not known. We design an algorithm that is guaranteed to find a third order local minimum.

Theorem 1 *(Informal) There is an algorithm that always converges to a third order local minimum (see Theorem 19). Also, in polynomial time the algorithm can find a point that is "similar" to a third order local minimum (see Theorem 18).*

By "similar" we mean the point x approximately satisfies the necessary and sufficient condition for third order local minimum (see Definition 9): the gradient $\nabla f(x)$ is small, Hessian $\nabla^2 f(x)$ is almost positive semidefinite (p.s.d) and in every subspace where the Hessian is small, the norm of the third order derivatives is also small.

To the best of our knowledge this is the first algorithm that is guaranteed to converge to a third order local minimum. The algorithm alternates between a second order step (which we use cubic regularization(Nesterov and Polyak, 2006)) and a third order step. The third order step first identifies a "competitive subspace" where the third order derivative has a much larger norm than the second order. It then tries to find a good direction in this subspace to make improvement. For more details see Section 5.

We also show that it is NP-hard to find a fourth order local minimum:

Theorem 2 *(Informal) Even for a well-behaved function, it is NP-hard to find a fourth order local minimum (see Theorem 24).*

1.2. Related Work

A popular approach to overcoming saddle points is to incorporate second order information. However, the popular second order approach of Newton's method is not suitable since it converges to an arbitrary critical point, and does not distinguish between a local minimum and a saddle point. Directions along negative values of the Hessian matrix help in escaping the saddle point. A simple solution is then to use these directions, whenever gradient descent improvements are small (which signals the approach towards a critical point) (Frieze et al., 1996; Vempala and Xiao, 2011).

A more elegant framework is the so-called trust region method (Dauphin et al., 2014; Sun et al., 2015) which involves optimizing the second order Taylor's approximation of the

objective function in a local neighborhood of the current point. Intuitively, this objective "switches" smoothly between first order and second order updates. Nesterov and Polyak (2006) propose adding a cubic regularization term to this Taylor's approximation. In a beautiful result, they show that in each step, this cubic regularized objective can be solved optimally due to hidden convexity and overall, the algorithm converges to a local optimum in bounded time. We give an overview of this algorithm in Section 3. Baes (2009) generalizes this idea to use higher order Taylor expansion, however the optimization problem is intractable even for third order Taylor expansion with quartic regularizer. Ge et al. (2015) recently showed that it is possible to escape saddle points using only first order information based on noisy stochastic gradient descent (SGD) in polynomial time in high dimensions. In many applications, this is far cheaper than the computation of the Hessian eigenvectors. Nie (2015) propose using the hierarchy of semi-definite relaxations to compute all the local optima which satisfy first and second order necessary conditions based on semi-definite relaxations.

All the above works deal with local optimality based on second order conditions. When the Hessian matrix is singular and p.s.d., higher order derivatives are required to determine whether it is a local optimum or a saddle point. Higher order optimality conditions, both necessary and sufficient, have been characterized before, e.g. (Bernstein, 1984; Warga, 1986). But these conditions are not efficiently computable, and it is NP-hard to determine local optimality, given such information about higher order derivatives (Murty and Kabadi, 1987).

2. Preliminaries

In this section we first introduce the classifications of saddle points. Next, as we often work with third order derivatives, and we treat it as a order 3 tensor, we introduce the necessary notations for tensors.

2.1. Critical Points

Throughout the paper we consider functions $f : \mathbb{R}^n \to \mathbb{R}$ whose first three order derivatives exist. We represent the derivatives by $\nabla f(x) \in \mathbb{R}^n$, $\nabla^2 f(x) \in \mathbb{R}^{n\times n}$ and $\nabla^3 f(x) \in \mathbb{R}^{n^3}$, where

$$[\nabla f(x)]_i = \frac{\partial}{\partial x_i} f(x), [\nabla^2 f(x)]_{i,j} = \frac{\partial^2}{\partial x_i \partial x_j} f(x), [\nabla^3 f(x)]_{i,j,k} = \frac{\partial^3}{\partial x_i \partial x_j \partial x_k} f(x).$$

For such smooth function $f(x)$, we say x is a *critical point* if $\nabla f(x) = \mathbf{0}$. Traditionally, critical points are classified into four cases according to the Hessian matrix:

1. (Local Minimum) All eigenvalues of $\nabla^2 f(x)$ are positive.
2. (Local Maximum) All eigenvalues of $\nabla^2 f(x)$ are negative.
3. (Strict saddle) $\nabla^2 f(x)$ has at least one positive and one negative eigenvalues.
4. (Degenerate) $\nabla^2 f(x)$ has either nonnegative or nonpositive eigenvalues, with some eigenvalues equal to 0.

As we shall see later in Section 3, for the first three cases second order algorithms can either find a direction to reduce the function value (in case of local maximum or strict

saddle), or correct asserting that the current point is a local minimum. However, second order algorithms cannot handle degenerate saddle points.

Degeneracy of Hessian indicates the presence of a *gutter* structure, where a set of connected points all have the same value, and all are local minima, maxima or saddle points (Dauphin et al., 2014). See for example Figure 1 (c) (d).

If the Hessian at a critical point x is p.s.d., even if it has 0 eigenvalues we can say the point is a second order local minimum: for any y that is sufficiently close to x, we have $f(x) - f(y) = o(\|x-y\|^2)$. That is, although there might be a vector y that makes the function value decrease, the amount of decrease is a lower order term compared to $\|x-y\|^2$. In this paper we consider higher order local minimum:

Definition 3 (p-th order local minimum) *A critical point x satisfies the p-th order necessary condition for local minimum (which we shorten as x is a p-th order local minimum throughout the paper), if there exists constants $C, \epsilon > 0$ such that for every y with $\|y - x\| \le \epsilon$,*

$$f(y) \ge f(x) - C\|x-y\|^{p+1}.$$

Every critical point is a first order local minimum, and every point that satisfies the second order necessary condition ($\nabla f(x) = 0, \nabla^2 f(x) \succeq 0$) is a second order local minimum.

2.2. Matrix and Tensor Notations

For a vector $v \in \mathbb{R}^n$, we use $\|v\|$ to denote its ℓ_2 norm. For a matrix $M \in \mathbb{R}^{n\times n}$, we use $\|M\|$ to denote its spectral (operator) norm. All the matrices we consider are symmetric matrices, and they can be decomposed using eigen-decomposition:

$$M = \sum_{i=1}^{n} \lambda_i v_i v_i^\top.$$

In this decomposition v_i's are orthonormal vectors, and λ_i's are the eigenvalues of M. We always assume $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_n$. We use $\lambda_1(M)$ to denote its largest eigenvalue and $\lambda_n(M)$ to denote its smallest eigenvalue. By the property of symmetric matrices we also know $\|M\| = \max\{|\lambda_1(M)|, |\lambda_n(M)|\}$. We use $\|M\|_F$ to denote the Frobenius norm of the matrix $\|M\|_F = \sqrt{\sum_{i,j\in[n]} M_{i,j}^2}$.

The third order derivative is represented by a $n \times n \times n$ tensor T. We use the following multilinear notation to simplify the notations of tensors:

Definition 4 (Multilinear notations) *Let $T \in \mathbb{R}^{n\times n\times n}$ be a third order tensor. Let $U \in \mathbb{R}^{n\times n_1}$, $V \in \mathbb{R}n \times n_2$ and $W \in \mathbb{R}^{n\times n_3}$ be three matrices, then the multilinear form $T(U,V,W)$ is a tensor in $\mathbb{R}^{n_1\otimes n_2\otimes n_3}$ that is equal to*

$$[T(U,V,W)]_{p,q,r} = \sum_{i,j,k\in[n]} T_{i,j,k} U_{i,p} V_{j,q} W_{k,r}.$$

In particular, for vectors $u, v, w \in \mathbb{R}^n$, $T(u,v,w)$ is a number that relates linearly in u, v and w (similar to $u^\top M v$ for a matrix); $T(u,v,I)$ is a vector in $\mathbb{R}^n$ (similar to Mu for a matrix); $T(u,I,I)$ is a matrix in $\mathbb{R}^{n\times n}$.

The Frobenius norm of a tensor T is defined similarly as matrices: $\|T\|_F = \sqrt{\sum_{i,j,k\in[n]} T_{i,j,k}^2}$. The spectral norm (also called injective norm) of a tensor is defined as

$$\|T\| = \max_{\|u\|=1,\|v\|=1,\|w\|=1} T(u,v,w).$$

We say a tensor is symmetric if $T_{i,j,k} = T_{\pi(i,j,k)}$ for any permutation of the indices. For symmetric tensors the spectral norm is also equal to $\|T\| = \max_{\|u\|=1} T(u,u,u)$. In both cases it is NP-hard to compute the spectral norm of a tensor(Hillar and Lim, 2013).

We will often need to project a tensor T to a subspace $\mathcal{P}$. Let P be the projection matrix to the subspace P, we use the notation $\text{Proj}_{\mathcal{P}} T$ which denotes $T(P,P,P)$. Intuitively, $[T(P,P,P)]_{u,v,w} = T(Pu,Pv,Pw)$, that is, the projected tensor applied to vector u,v,w is equivalent to the original tensor applied to the projection of u,v,w.

3. Overview of Nestorov's Cubic Regularization

In this section we review the guarantees of Nesterov's Cubic Regularization algorithm(Nesterov and Polyak, 2006). We will use this algorithm as a key step later in Section 5, and prove analogous results for third order local minimum. Note that we can replace Cubic Regularization with other second order algorithms such as trust region algorithms, we use Cubic Regularization mostly to simplify the proof.

The algorithm requires the first two order derivatives exist and the following smoothness constraint:

Assumption 1 (Lipschitz-Hessian)

$$\forall x,y, \|\nabla^2 f(x) - \nabla^2 f(y)\| \le R\|x-y\|.$$

At a point x, the algorithm tries to find a nearby point z that optimizes the degree two Taylor's expansion: $f(x)+\langle\nabla f(x), z-x\rangle+\frac{1}{2}(z-x)^\top(\nabla^2 f(x))(z-x)$, with the cubic distance $\frac{R}{6}\|z-x\|^3$ as a regularizer. See Algorithm 1 for one iteration of the algorithm. The final algorithm generates a sequence of points $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ where $x^{(i+1)} = \text{CubicReg}(x^{(i)})$.

Algorithm 1 CubicReg(Nesterov and Polyak, 2006)

Require: function f, current point x, Hessian smoothness R
Ensure: Next point z that satisfies Theorem 6.
Let $z = \arg\min f(x) + \langle\nabla f(x), z-x\rangle + \frac{1}{2}(z-x)^\top(\nabla^2 f(x))(z-x) + \frac{R}{6}\|z-x\|^3$.
return z

The optimization problem that Algorithm 1 tries to solve may seem difficult, as it has a cubic regularizer $\|z-x\|^3$. However, Nesterov and Polyak (2006) showed that it is possible to solve this optimization problem in polynomial time.

For each point, define $\mu(z)$ to measure how close the point z is to satisfying the second order optimality condition:

Definition 5 $\mu(z) = \max\left\{\sqrt{\frac{1}{R}\|\nabla f(z)\|}, -\frac{2}{3R}\lambda_n\nabla^2 f(z)\right\}$

When $\mu(z) = 0$ we know $\nabla f(z) = 0$ and $\nabla^2 f(z) \succeq 0$, which satisfies the second order necessary conditions (and in fact implies that z is a second order local minimum). When $\mu(z)$ is small we can say that the point z approximately satisfies the second order optimality condition.

For one step of the algorithm the following guarantees can be proven[1]

Theorem 6 *(Nesterov and Polyak, 2006) Suppose $z = CubicRegularize(x)$, then $\|z-x\| \geq \mu(z)$ and $f(z) \leq f(x) - R\|z-x\|^3/12$.*

Using Theorem 6, Nesterov and Polyak (2006) can get strong convergence results for the sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$

Theorem 7 *(Nesterov and Polyak, 2006) If $f(x)$ is bounded below by $f(x^*)$, then $\lim_{i\to\infty} \mu(x^{(i)}) = 0$, and for any $t \geq 1$ we have*

$$\min_{1\leq i\leq t} \mu(x^{(i)}) \leq \frac{8}{3} \cdot \left(\frac{3(f(x^{(0)}) - f(x^*))}{2tR}\right)^{1/3}.$$

This theorem shows that within first t iterations, we can find a point that "looks similar" to a second order local minimum in the sense that gradient is small and Hessian does not have a negative eigenvalue with large absolute value. It is also possible to prove stronger guarantees for the limit points of the sequence:

Theorem 8 *(Nesterov and Polyak, 2006) If the level set $\mathcal{L}(x^{(0)}) := \{x | f(x) \leq f(x^{(0)})\}$ is bounded, then the following limit exists*

$$\lim_{i\to\infty} f(x^{(i)}) = f^*,$$

The set X^* of the limit points of this sequence is non-empty. Moreover this is a connected set such that for any $x \in X^*$ we have

$$f(x) = f^*, \nabla f(x) = \mathbf{0}, \nabla^2 f(x) \succeq 0.$$

Therefore the algorithm always converges to a set of points that are all second order local minima.

4. Third Order Necessary Condition

In this section we present a condition for a point to be a third order local minimum, and show that it is necessary and sufficient for a class of smooth functions. Proofs are deferred to Appendix B.1.

All the functions we consider satisfies the following natural smoothness conditions

1. All of guarantees we stated here correspond to setting the regularizer R to be exactly equal to the smoothness in Assumption 1.

Assumption 2 (Lipschitz third Order) *We assume the first three derivatives of $f(x)$ exist, and for any $x, y \in \mathbb{R}^n$,*

$$\|\nabla^3 f(x) - \nabla^3 f(y)\|_F \leq L\|x - y\|.$$

Under this assumption, we state our conditions for a point to be a third order local minimum.

Definition 9 (Third-order necessary condition) *A point x satisfy third-order necessary condition, if*

1. $\nabla f(x) = 0$.
2. $\nabla^2 f(x) \succeq 0$.
3. *For any u that satisfy $u^\top(\nabla^2 f(x))u = 0$, $[\nabla^3 f(x)](u, u, u) = 0$.*

We first note that this condition can be verified in polynomial time.

Claim 1 *Conditions in Definition 9 can be verified in polynomial time given the gradients $\nabla f(x), \nabla^2 f(x)$ and $\nabla^3 f(x)$.*

Proof It is easy to check whether $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$. We can also use SVD to compute the subspace $\mathcal{P}$ such that $u^\top(\nabla^2 f(x))u = 0$ if and only if $u \in \mathcal{P}$.

Now we can compute the projection of $\nabla^3 f(x)$ in the subspace $\mathcal{P}$, and we claim the third condition is violated if and only if the projection is nonzero.

If the projection is zero, then clearly $[\nabla^3 f(x)](u, u, u)$ is 0 for any $u \in \mathcal{P}$. On the other hand, if projection Z is nonzero, let u be a uniform Gaussian vector that has unit variance in all directions of u, then we know $\mathbb{E}[[[\nabla^3 f(x)](u, u, u)]^2] \geq \|Z\|_F^2 > 0$, so there must exists an $u \in \mathcal{P}$ such that $[\nabla^3 f(x)](u, u, u) \neq 0$. ■

Theorem 10 *Given a function f that satisfies Assumption 2, a point x is third order optimal if and only if it satisfies Condition 9.*

Before proving the theorem, we first show a bound on $f(y)$ and a Taylor's expansion of f at point x.

Lemma 11 *For any x, y, we have*

$$|f(y)-f(x)-\langle\nabla f(x), y-x\rangle+\frac{1}{2}(y-x)^\top\nabla^2 f(x)(y-x)-\frac{1}{6}\nabla^3 f(x)(y-x, y-x, y-x)| \leq \frac{L}{24}\|y-x\|^4.$$

The Lemma can be proved by integrating over the third order derivatives three times and bounding the differences. Details are deferred to Appendix B.1.

This lemmas allow us to ignore the fourth order term $\|y - x\|^4$ and focus on the order 3 Taylor expansion when $\|y - x\|$ is small. To prove Theorem 10, intuitively, the "only if" direction (local minimum to necessary condition) is easy because if any condition in Definition 9 is violated, we can use that particular derivative to find a direction that improves the function value. For the "if" direction (necessary condition to third order local minimum), the main challenge is to balance the contribution we get from the positive part of the Hessian matrix and the third order derivatives. For details see Appendix B.1.

5. Algorithm for Finding Third Order Optimal Points

We design an algorithm that is guaranteed to converge to a third order local minimum. Throughout this section we assume both Assumptions 1 and 2 [2].

The main intuition of the algorithm is similar to the proof of Theorem 10: the algorithm tries to make improvements using first, second or third order information. However, the nature of the third order condition makes it challenging for the algorithm to guarantee progress.

Consider a potential local minimum point x. It is very easy to check whether $\nabla f(x) \neq 0$ or $\lambda_{min}(\nabla^2 f(x)) < 0$, and to make progress using the corresponding directions. However, to verify Condition 3 in Definition 9, we need to do it in the right subspace.

At the critical point, Condition 3 refers to a particular subspace which is the nullspace of the Hessian matrix. However, if we consider a point y that is very close to this critical point x, no matter how close they are the Hessian of y may still be positive definite (although it must have really small eigenvalues near the nullspace of $\nabla^2 f(x)$). The point y will appear to satisfy Condition 3 in Definition 9 and the algorithm might incorrectly think y is close to a local minimum, while the truth is y is only close to a saddle point. We do not want to the algorithm to spend too much time around such points , so we need to identify a subspace that may have some positive eigenvalues. In order to make sure we can find a vector the contribution from third order term is larger than the second order term. Based on this intuition we define *competitive subspace* below:

Definition 12 (eigensubspace) *For any symmetric matrix M, let its eigendecomposition be $M = \sum_{i=1}^{n} \lambda_i v_i v_i^\top$ (where λ_i's are eigenvalues and $\|v_i\| = 1$), we use $\mathcal{S}_\tau(M)$ to denote the span of eigenvectors with eigenvalue at most τ. That is*

$$\mathcal{S}_\tau(M) = span\{v_i | \lambda_i \leq \tau\}.$$

Definition 13 (competitive subspace) *For any $Q > 0$, and any point z, let the competitive subspace $\mathcal{S}(z)$ be the largest eigensubspace $\mathcal{S}_\tau(\nabla^2 f(z))$, such that if we let $C_Q(z)$ be the norm of the third order derivatives in this subspace*

$$C_Q(z) = \|Proj_{\mathcal{S}(z)} \nabla^3 f(z)\|_F,$$

then $\tau \leq C_Q^2 / 12LQ^2$.

If no such subspace exists then let $\mathcal{S}(z)$ be empty and $C_Q(z) = 0$.

Similar to $\mu(z)$ as in Definition 5, $C_Q(z)$ can be viewed as how Condition 3 in Definition 9 is satisfied approximately. If both $\mu(z)$ and $C_Q(z)$ are 0 then the point z satisfies third order necessary conditions.

Intuitively, competitive subspace is a subspace where the eigenvalues of the Hessian are small, but the Frobenius norm of the third order derivative is large. Therefore we are likely to make progress using the third order information. The parameters in Definition 13 are set so that if there is a unit vector $u \in \mathcal{S}(z)$ such that $[\nabla^3 f(z)](u,u,u) \geq \|\text{Proj}_{\mathcal{S}(z)} \nabla^3 f(z)\|_F / Q$ (see Theorem 16), then we can find a new point where the sum of second, third and fourth order term can be bounded (see Lemma 17).

2. Note that we actually only cares about a *level set* $\mathcal{L} = \{x | f(x) \leq f(x^{(0)})\}$, as long as this set is bounded Assumptions 1 follows from Assumption 2

Remark 14 *The competitive subspace in Definition 13 can be computed in polynomial time, see Algorithm 4. The main idea is that we can compute the eigendecomposition of the Hessian* $\nabla^2 f(z) = \sum_{i=1}^n \lambda_i v_i v_i^\top$*, and then there are only* n *different subspaces (span*$\{v_n\}$*, span*$\{v_{n-1}, v_n\}$*, ...,* $span\{v_1, v_2, \dots v_n\}$*). We can enumerate over all of them, and check for which subspaces the norm of the third order derivative is large.*

Now we are ready to state the algorithm. The algorithm is a combination of the cubic regularization algorithm and a third order step that tries to use the third order derivative in order to improve the function value in the competitive subspace.

Remark 15 *Note that the algorithm is stated in a way to simplify the proof and the third order step may be slow in practice. In practice, we can run the cubic regularization algorithm, and only apply the third order step when cubic regularization is not making enough progress. This will make the algorithm always at least as fast as cubic regularization.*

Algorithm 2 Third Order Optimization

for $i = 0$ **to** $t - 1$ **do**
  $z^{(i)} = \text{CubicReg}(x^{(i)})$.
  Let $\epsilon_1 = \|\nabla f(z^{(i)})\|$,
  Let $\mathcal{S}(z), C_Q(z)$ be the competitive subspace of $f(z)$ (Definition 13).
  if $C_Q(z) \geq Q(24\epsilon_1 L)^{1/3}$ **then**
    $u = \text{Approx}(\nabla^3 f(z^{(i)}), \mathcal{S})$.
    $x^{(i+1)} = z^{(i)} - \frac{C_Q(z)}{LQ} u$.
  else
    $x^{(i+1)} = z^{(i)}$.
  end if
end for

Suppose we have the following approximation guarantee for Algorithm 3

Algorithm 3 Approximate Tensor Norms

Require: Tensor T, subspace $\mathcal{S}$.
Ensure: unit vector $u \in \mathcal{S}$ such that $T(u, u, u) \geq \|\text{Proj}_{\mathcal{S}} T\|_F / Q$.
  repeat
    Let $\hat{u}$ be a random standard Gaussian in subspace $\mathcal{S}$.
    Let $u = \hat{u}$
  until $|T(u, u, u)| \geq \|\text{Proj}_{\mathcal{S}} T\|_F / Bn^{1.5}$ for a fixed constant B
  return u if $T(u, u, u) > 0$ and $-u$ otherwise.

Theorem 16 *There is a universal constant* B *such that the expected number of iterations of Algorithm 3 is at most 2, and the output of Approx is a unit vector* u *that satisfies* $T(u, u, u) \geq \|Proj_{\mathcal{S}} T\|_F / Q$ *for* $Q = Bn^{1.5}$.

The proof of this theorem follows directly from anti-concentration (see Appendix B.2. Notice that there are other algorithms that can potentially give better approximation (lower value of Q) which will improve the rate of our algorithm. However in this paper we do not try to optimize over dependencies over the dimension n, that is left as an open problem.

By the choice of the parameters in the algorithm, we can get the following guarantee (which is analogous to Theorem 6):

Lemma 17 *If $C_Q(z) \geq Q(24\epsilon_1 L)^{1/3}$, u is a unit vector in $\mathcal{S}(z)$ and $[\nabla^3 f(z)](u,u,u) \geq \|Proj_{\mathcal{S}(z)}\nabla^3 f(z)\|_F/Q$. Let $x' = z - C_Q(z)/LQ \cdot u$. then we have*

$$f(x') \leq f(z) - \frac{C_Q(z)^4}{24L^3Q^4}.$$

Proof Let $\epsilon = C_Q(z)/LQ$, then by Lemma 11 we know

$$f(x') \leq f(z) - \frac{\epsilon^3 C}{6Q} + \epsilon_1\epsilon + \epsilon_2\epsilon^2/2 + L\epsilon^4/24.$$

Here $\epsilon_1 = \|\nabla f(z)\|$, and $\epsilon_2 \leq \frac{C_Q(z)^2}{12LQ^2}$ by the construction of the subspace.

By the choice of parameters, we know the terms $\epsilon_1\epsilon, \epsilon_2\epsilon^2/2, L\epsilon^4/24$ are all bounded by $\frac{\epsilon^3 C_Q(z)}{24Q}$, therefore

$$f(x') \leq f(z) - \frac{\epsilon^3 C_Q(z)}{24Q} = f(z) - \frac{C_Q(z)^4}{24L^3Q^4}$$

■

Using this Lemma, and Theorem 6 for cubic regularization, we can show that both progress measure goes to 0 as the number of steps increase (this is analogous to Theorem 7).

Theorem 18 *Suppose the algorithm starts at $f(x_0)$, and f has global min at $f(x^*)$. Then in one of the t iterations we have*

1. $\mu(z) \leq \left(\frac{12(f(x_0)-f(x^*)}{Rt}\right)^{1/3}$.
2. $C_Q(z) \leq \max\left\{Q(24\|\nabla f(z)\|L)^{1/3}, Q\left(\frac{24L^3(f(x_0)-f(x^*))}{t}\right)^{1/4}\right\}$.

Recall $\mu(z) = \max\left\{\sqrt{\frac{1}{R}\|\nabla f(z)\|}, -\frac{2}{3R}\lambda_n\nabla^2 f(z)\right\}$ is intuitively measuring how much first and second order progress the algorithm can make. The value $C_Q(z)$ as defined in Definition 13 is a measure of how much third order progress the algorithm can make. The theorem shows both values goes to 0 as t increases (note that even the first term $Q(24\|\nabla f(z)\|L)^{1/3}$ in the bound for $C_Q(z)$ goes to 0 because the $\|\nabla f(z)\|$ goes to 0).

Proof By the guarantees of Theorem 6 and Lemma 17, we know the sequence of points $x^{(0)}, z^{(0)}, \ldots, x^{(i)}, z^{(i)}, \ldots$ has non-increasing function values. Also,

$$\sum_{i=1}^{t} f(x^{(i)}) - f(x^{(i-1)}) \leq f(x_0) - f(x^*).$$

So there must be an iteration where $f(x^{(i)}) - f(x^{(i-1)}) \leq \frac{f(x_0)-f(x^*)}{t}$.

If $\mu(z) > \left(\frac{12(f(x_0)-f(x^*)}{Rt}\right)^{1/3}$, then Theorem 6 implies $f(x^{(i-1)}) - f(z^{(i-1)}) > \frac{f(x_0)-f(x^*)}{t}$, which is impossible.

On the other hand if $C_Q(z) \geq \max\left\{Q(24\|\nabla f(z)\|L)^{1/3}, Q\left(\frac{24L^3(f(x_0)-f(x^*))}{t}\right)^{1/4}\right\}$, then the third order step makes progress, and we know $f(z^{(i-1)}) - f(x^{(i)}) > \frac{f(x_0)-f(x^*)}{t}$, which is again impossible. ■

We can also show that when t goes to infinity the algorithm converges to a third order local minimum (similar to Theorem 8).

Theorem 19 *When t goes to infinity, the values $f(x^{(t)})$ converge. If the level set $\mathcal{L}(f(x_0)) = \{x|f(x) \leq f(x_0)\}$ is compact, then the sequence of points $x^{(t)}, z^{(t)}$ has nonempty limit points, and every limit point x satisfies the third order necessary conditions.*

Proof By Theorem 6 and Lemma 17, we know the function value is non-increasing, and it has a lowerbound $f(x^*)$, so the value must converge.

The existence of limit points is guaranteed by the compactness of the level set. The only thing left to prove is that every limit point x must satisfy the third order necessary conditions.

Notice that $f(x^{(0)}) - \lim_{t\to\infty} f(x^{(t)} \geq \sum_{i=0}^{\infty} \frac{R\mu(z^{(i)})^3}{12} + \frac{C_Q(z^{(i)})^4}{24L^3Q^4}$, so $\lim_{i\to\infty} \mu(z^{(i)} = 0$ and $\lim_{i\to\infty} C_Q(z^{(i)}) = 0$. Also we know further $\lim_{i\to\infty} \|z^{(i)} - x^{(i)}\| = 0$. Therefore wlog a limit point x is also a limit point of sequence z, and $\lim_{i\to\infty} \|\nabla f(z)\| = 0$. Also we know $H = \nabla^2 f(x)$ is PSD, because otherwise points near x will have nonzero $\mu(z^{(i)}$ and x cannot be a limit point.

Now we only need to check the third order condition. Assume towards contradiction that third order condition is not true. Then we know the Hessian has a subspace $\mathcal{P}$ with 0 eigenvalues, and the third order derivative has norm at least ϵ in this subspace. By matrix perturbation theory, when z is very close to x, $\mathcal{P}$ is very close to $\mathcal{S}_\epsilon(z)$ for $\epsilon \to 0$. On the other hand, the third order tensor also converges to $\nabla^3 f(x)$ (by Lipschitz condition). Thus $\mathcal{S}_\epsilon(z)$ will eventually be a competitive subspace and $C_Q(z)$ is at least $\epsilon/2$ for all z. However this is impossible as $\lim_{i\to\infty} C_Q(z^{(i)}) = 0$. ■

Remark 20 *Note that not all third order local minimum can be the limit point for Algorithm 2. This is because if $f(x)$ has very large third order derivatives but relatively smaller Hessian, even though the Hessian might be positive definite (so x is in fact a local minimum), Algorithm 2 may still find a non-empty competitive subspace, and will be able to reduce the function value and escape from the saddle point. An example is for the function $f(x) = x^2 - 100x^3 + x^4$, $x = 0$ is a local minimum but the algorithm can escape from that and find the global minimum.*

In the most general case it is hard to get a convergence rate for the algorithm because the function may have higher order local minima. However, if the function has nice properties then it is possible to prove polynomial rates of convergence.

Definition 21 (strict third order saddle) *We say a function is strict third order saddle, if there exists constants $\alpha, c_1, c_2, c_3, c_4 > 0$ such that for any point x one of the following is true:*

1. $\|\nabla f(x)\| \geq c_1$.
2. $\lambda_n(f(x)) \leq -c_2$.
3. $C_Q(f(x)) \geq c_3$.
4. *There is a local minimum x^* such that $\|x - x^*\| \leq c_4$ and the function is α-strongly convex restricted to the region $\{x | \|x - x^*\| \leq 2c_4\}$.*

This is a generalization of the strict saddle functions defined in Ge et al. (2015). Even if a function has degenerate saddle points, it may still satisfy this condition.

Corollary 22 *When $t \geq$ poly$(n, L, R, Q, f(x_0) - f(x^*)) \max\{(1/c_1)^{1.5}, (1/c_2)^3, (1/c_3)^{4.5}\}$, there must be a point $z^{(i)}$ with $i \leq t$ that is in case 4 in Definition 21.*

Proof We use $\tilde{O}$ to only focus on the polynomial dependency on t and ignore polynomial dependency on all other parameters.

By Theorem 18, we know there must be a $z^{(i)}$ which satisfies $\mu(z^{(i)} \leq \tilde{O}((1/t)^{1/3})$ and $C_Q(z) \leq \tilde{O}(\max\{(1/t)^{1/4}, \|\nabla f(z)\|^{1/3}\})$.

By the Definition of μ (Definition 5), we know $\|\nabla f(z)\| \leq \tilde{O}(\mu(z))^2 = \tilde{O}(t^{-2/3})$, $\lambda_n(\nabla^2 f(z)) \geq -\tilde{O}(t^{-1/3})$.

Using the fact that $\|\nabla f(z)\| \leq \tilde{O}(\mu(z))^2 = \tilde{O}(t^{-2/3}$, we know

$$C_Q(z) \leq \tilde{O}(\max\{(1/t)^{1/4}, \|\nabla f(z)\|^{1/3}\}) = \tilde{O}(t^{-2/9}).$$

Therefore, when $t \geq$ poly$(n, L, R, Q, f(x_0) - f(x^*)) \max\{(1/c_1)^{1.5}, (1/c_2)^3, (1/c_3)^{4.5}\}$, the point z must satisfy

1. $\|\nabla f(z)\| < c_1$;
2. $\lambda_n(\nabla^2 f(z)) < -c_2$;
3. $C_Q(z) < c_3$.

Therefore the first three cases in Definition 21 cannot happen and z must be near a local minimum. ■

6. Hardness for Finding a fourth order Local Minimum

In this section we show it is hard to find a fourth order local minimum even if the function we consider is very well-behaved.

Definition 23 (Well-behaved function) *We say a function f is well-behaved if it is infinite-order differentiable, and satisfies:*

1. *$f(x)$ has a global minimizer at some point $\|x\| \leq 1$.*

2. *$f(x)$ has bounded first 5 derivatives for $\|x\| \le 1$.*
3. *For any direction $\|x\| = 1$, $f(tx)$ is increasing for $t \ge 1$.*

Clearly, all local minimizers of a well-behaved function lies within the unit ℓ_2 ball, and $f(x)$ is smooth with bounded derivatives within the unit ℓ_2 ball. These functions also satisfy Assumptions 1 and 2. All the algorithms mentioned in previous sections can work in this case and find a local minimum up to order 3. However, this is not possible for fourth order.

Theorem 24 *It is NP-hard to find a fourth order local minimum of a function $f(x)$, even if f is guaranteed to be well-behaved.*

The main idea of the proof comes from the fact that we cannot even verify the nonnegativeness of a degree 4 polynomial (hence there are cases where we cannot verify whether a point is a fourth order local minimum or not).

Theorem 25 *Nesterov (2000); Hillar and Lim (2013) It is NP-hard to tell whether a degree 4 homogeneous polynomial $f(x)$ is nonnegative.*

Remark 26 *The NP hardness for nonnegativeness of degree 4 polynomial has been proved in several ways. In Nesterov (2000) the reduction is from the SUBSET SUM problem, which results in a polynomial that can have exponentially large coefficients and does not rule out FPTAS. However, the reduction in Hillar and Lim (2013) relies on the hardness of copositive matrices, which in turn depends on the hardness of INDEPENDENT SET(Dickinson and Gijben, 2014). This reduction gives a polynomial whose coefficients can be bounded by poly(n), and a polynomial gap that rules out FPTAS. More precisely, this will rule out the possibility of finding a point x such that for nearby y,*

$$f(y) \ge f(x) - C\|x - y\|^4,$$

where C is a inverse polynomial factor. So finding a 4-th order local min is hard even in this approximation sense.

To prove Theorem 24 we only need to reduce the nonnegativeness problem in Theorem 25 to the problem of finding a fourth order local minimum. We can convert a degree 4 polynomial to a well behaved function by adding a degree 6 regularizer $\|x\|^6$. We shall show when the degree 4 polynomial is nonnegative the $\mathbf{0}$ point is the only fourth order local minimum; when the degree 4 polynomial has negative directions then every fourth order local minimum must have negative function value. The details are deferred to Section B.3.

7. Conclusion

Complicated structures of saddle points are a major problem for optimization algorithms. In this paper we investigate the possibilities of using higher order derivatives in order to avoid degenerate saddle points. We give the first algorithm that is guaranteed to find a 3rd order local minimum, which can solve some problems caused by degenerate saddle points. However, we also show that the same ideas cannot be easily generalized to higher orders.

There are still many open problems related to degenerate saddle points and higher order optimization algorithms. Are there interesting class of functions that satisfies the strict 3rd order saddle property (Definition 21)? Can simple algorithms such as stochastic gradient descent escape higher order saddle functions in polynomial time? Can we design a 3rd order optimization algorithm for constrained optimization? We hope this paper inspires more research in these directions and eventually design efficient optimization algorithms whose performance do not suffer from degenerate saddle points.

Acknowledgements

Animashree Anandkumar is supported in part by Microsoft Faculty Fellowship, NSF Career Award CCF- 1254106, ONR Award N00014-14-1-0665, ARO YIP Award W911NF-13-1-0084, and AFOSR YIP FA9550- 15-1-0221.

References

Shun-Ichi Amari, Hyeyoung Park, and Tomoko Ozeki. Singularities affect dynamics of learning in neuromanifolds. *Neural computation*, 18(5):1007–1065, 2006.

Peter Auer, Mark Herbster, Manfred K Warmuth, et al. Exponentially many local minima for single neurons. *Advances in neural information processing systems*, pages 316–322, 1996.

Antonio Auffinger, Gerard Ben Arous, et al. Complexity of random smooth functions on the high-dimensional sphere. *The Annals of Probability*, 41(6):4214–4247, 2013.

Michel Baes. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2009.

Dennis S Bernstein. A systematic approach to higher-order necessary conditions in optimization theory. *SIAM journal on control and optimization*, 22(2):211–238, 1984.

Anthony Carbery and James Wright. Distributional and l^ q norm inequalities for polynomials over convex bodies in r^ n. *Mathematical Research Letters*, 8(3):233–248, 2001.

Dustin Cartwright and Bernd Sturmfels. The number of eigenvalues of a tensor. *Linear algebra and its applications*, 438(2):942–952, 2013.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

Peter JC Dickinson and Luuk Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Computational optimization and applications*, 57(2):403–415, 2014.

Alan Frieze, Mark Jerrum, and Ravi Kannan. Learning linear transformations. In *focs*, page 359. IEEE, 1996.

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.

Rich Caruana Steve Lawrence Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, volume 13, page 402, 2001.

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

Masato Inoue, Hyeyoung Park, and Masato Okada. On-line learning theory of soft committee machines with correlated hidden units–steepest gradient descent and natural gradient descent–. *Journal of the Physical Society of Japan*, 72(4):805–810, 2003.

Katta G Murty and Santosh N Kabadi. Some np-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129, 1987.

Yurii Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, pages 405–440. Springer, 2000.

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Jiawang Nie. The hierarchy of local minimums in polynomial optimization. *Mathematical Programming*, 151(2):555–583, 2015.

David Saad and Sara A Solla. On-line learning in soft committee machines. *Physical Review E*, 52(4):4225, 1995.

Itay Safran and Ohad Shamir. On the quality of the initial basin in overspecified neural networks. *arXiv preprint arXiv:1511.04210*, 2015.

Ju Sun, Qing Qu, and John Wright. When are nonconvex problems not scary? *arXiv preprint arXiv:1510.06096*, 2015.

Santosh S Vempala and Ying Xiao. Structure from local optima: Learning subspace juntas via higher order pca. *arXiv preprint arXiv:1108.3329*, 2011.

Jack Warga. Higher order conditions with and without lagrange multipliers. *SIAM journal on control and optimization*, 24(4):715–730, 1986.

Haikun Wei, Jun Zhang, Florent Cousseau, Tomoko Ozeki, and Shun-ichi Amari. Dynamics of learning near singularities in layered networks. *Neural computation*, 20(3):813–843, 2008.

Appendix A. Degree 3 Polynomials and Third Order Conditions

Since the third order optimal points depend on the first three derivatives of the objective function, it might be tempting to conjecture that for a degree 3 polynomial, a third order local minimum is also a true local minimum. While this is in some sense true, optimizing a degree 3 polynomial without any constraint is not a very interesting problem, because either the polynomial is always equal to 0 or the answer is infinity. The problem becomes more interesting when we add constraints. For example, let $p(x)$ be a degree 3 homogeneous polynomial, then we might want to optimize:

$$\begin{aligned} \min \quad & p(x) \\ s.t. \quad & \|x\| = 1. \end{aligned} \tag{1}$$

As we mentioned before, this is exactly the problem of computing the spectral norm of a symmetric tensor.

When the corresponding tensor is orthogonal (which means there are orthonormal vectors $v_1, v_2, ..., v_n$ such that $p(x) = \sum_{i=1}^n \lambda_i \langle v_i, x\rangle^3$), previous works (see e.g.Ge et al. (2015)) showed that second order conditions are enough to ensure we find a local minimum for this function, and all local minima are aligned with one of the components $v_1, ..., v_n$.

However, the above result is not true for general cubic polynomials on the sphere.

Claim 2 *There is a degree 3 polynomial $p(x)$ and a point x, such that x is a third order local minimum for optimization problem (1) or (2), but it is not a local minimum.*

Proof Consider the function $p(x) = x_1^3 + 1.5x_1x_2^2 + 100x_2^2x_3$ and the point $x = (-1, 0, 0)$. It is not very hard to check that x is indeed a third order local minimum (when the problem is constrained, we also only consider y such that $\|y\| = 1$). In particular for Objective (2), we can see that the third order condition is satisfied: the gradient is equal to 0, Hessian is PSD with the only degenerate direction being $(0, 1, 0)$, and the projection of the third order derivative in that direction is 0 (there is no x_2^3 term).

However, this point is not a true local minimum, because we can consider points $y_\epsilon = (-\sqrt{1-\epsilon^2-\epsilon^4}, \epsilon, -\epsilon^2)$. When ϵ goes to 0, these points can become arbitrarily close to x, however the objective value $p(y_\epsilon) = p(x) - \Theta(\epsilon^4)$. ■

We can convert the constrained cubic polynomial on the sphere in (1) to an unconstrained quartic polynomial through regularization

$$\min p(x) + \frac{3}{4}\|x\|^4. \tag{2}$$

However, as we noted in Theorem 25, obtaining the local optima of a quartic polynomial is NP-hard.

Appendix B. Omitted Proofs

B.1. Omitted Proofs in Section 4

Lemma 27 (Lemma 11 Restated) *For any x, y, we have*

$$|f(y)-f(x)-\langle\nabla f(x), y-x\rangle+\frac{1}{2}(y-x)^\top\nabla^2 f(x)(y-x)-\frac{1}{6}\nabla^3 f(x)(y-x, y-x, y-x)| \le \frac{L}{24}\|y-x\|^4.$$

Proof The proof follows from integration from x to y repeatedly.

First we have

$$\nabla^2 f(x+u(y-x)) = \nabla^2 f(x) + \left[\int_0^u \nabla^3 f(x+v(y-x))dv\right](y-x).$$

By the Lipschitz condition on third order derivative, we know

$$\|\nabla^3 f(x+v(y-x)) - \nabla^3 f(x)\|_F \leq Lv\|x-y\|.$$

Combining the two we have

$$\nabla^2 f(x+u(y-x)) = \nabla^2 f(x) + [\nabla^3 f(x)](y-x) + h(u),$$

where $h(u) = \left[\int_0^u (\nabla^3 f(x+v(y-x)) - \nabla^3 f(x))dv\right](y-x)$, so $\|h(u)\|_F \leq \frac{L}{2}\|x-y\|^2$.

Now we use the integral for the gradient of f:

$$\begin{aligned}\nabla f(x+t(y-x)) &= \nabla f(x) + \left[\int_0^t \nabla^2 f(x+u(y-x))du\right](y-x)\\ &= \nabla f(x) + \nabla^2 f(x)(y-x) + \left[\int_0^t h(u)du\right](y-x).\end{aligned}$$

Let $g(t) = \left[\int_0^t h(u)du\right](y-x)$, by the bound on $h(u)$ we know $\|g(t)\| \leq \frac{1}{6}\|x-y\|^3$. Finally, we have

$$\begin{aligned}f(y) &= f(x) + \langle\left[\int_0^1 \nabla f(x+t(y-x))du\right], (y-x)\rangle\\ &= f(x) + \langle\nabla f(x), y-x\rangle + \frac{1}{2}(y-x)^\top\nabla^2 f(x)(y-x) + \frac{1}{6}\nabla^3 f(x)(y-x)^{\otimes 3} + \langle\left[\int_0^1 g(t)dt\right], y-x\rangle.\end{aligned}$$

The last term is bounded by $\|y-x\|\int_0^1\|g(t)\|dt \leq \frac{L}{24}\|x-y\|^4$. ■

Theorem 28 (Theorem 10 restated) *Given a function f that satisfies Assumption 2, a point x is third order optimal if and only if it satisfies Condition 9.*

Proof (necessary condition → third order minimal) By Lemma 11 we know

$$f(y) \geq f(x) + \langle\nabla f(x), y-x\rangle + \frac{1}{2}(y-x)^\top\nabla^2 f(x)(y-x) + \frac{1}{6}\nabla^3 f(x)(y-x)^{\otimes 3} - \frac{L}{24}\|y-x\|^4.$$

Now let α be the smallest nonzero eigenvalue of $\nabla^2 f(x)$. Let U be nullspace of $\nabla^2 f(x)$ and V be the orthogonal subspace. We break $\nabla^3 f(x)$ into two tensors G_1 and G_2, where G_1 is the projection to $V\otimes V\otimes V$, $V\otimes V\otimes U$ (and its symmetries), and G_2 is the projection to $V\otimes U\otimes U$ (and its symmetries). Note that $\nabla^3 f(x) = G_1 + G_2$ because the projection on $U\otimes U\otimes U$ is 0 by the third condition. Let β be the max injective norm of G_1 and G_2.

Now we know for any $u\in U$ and $v\in V$,

$$f(x+u+v) - f(x) \geq \frac{1}{2}\alpha\|v\|^2 - \frac{\beta}{6}\|u\|\|v\|^2 - \frac{\beta}{6}\|u\|^2\|v\| - \frac{L}{24}\|u+v\|^4.$$

Now, if $\epsilon < \beta/\alpha$, because $\|u\|_2 \le \epsilon$ it is easy to see the sum of first two terms is at least $\frac{1}{3}\alpha\|v\|_2^2$. Now we can take the mininum of

$$\frac{\alpha}{3}\|v\|^2 - \frac{\beta}{6}\|u\|^2\|v\|,$$

The minimum is achieved when $\|v\| = \|u\|^2\beta/\alpha$ and the minimum value is $-\|u\|^4\beta^2/6\alpha$. Therefore when $\|u+v\| \le \beta/\alpha$ we have

$$f(x+u+v) - f(x) \ge -\left(\frac{\beta^2}{\alpha} + \frac{L}{24}\right)\|u+v\|^4.$$

(third order minimal→necessary condition) Assume towards contradiction that the necessary condition is not satisfied, but the point x is third order local optimal.

If the necessary condition is not satisfied, then one of the three cases happens:

In the first case the gradient $\nabla f(x) \neq 0$. In this case, if we let L' be an upperbound the operator norms of the second and third order derivative, then we know

$$f(x+\epsilon\nabla f(x)) \le f(x) - \epsilon\|\nabla f(x)\|^2 + \frac{\epsilon^2 L'}{2}\|\nabla f(x)\|^2 + \frac{\epsilon^3 L'}{6}\|\nabla f(x)\|^3 + \frac{\epsilon^4 L}{24}\|\nabla f(x)\|^4.$$

When $\epsilon\|\nabla f(x)\| \le 1$ and $\epsilon(2L'/3 + L/24) \le 1/2$, we have

$$f(x+\epsilon\nabla f(x)) \le f(x) - \frac{\epsilon}{2}\|\nabla f(x)\|^2.$$

Therefore the point cannot be a third order local minimum.

In the second case, $\nabla f(x) = 0$, but $\lambda_{min}\nabla^2 f(x) < 0$. Let $\|u\| = 1$ be a unit vector such that $u^\top(\nabla^2 f(x))u = -c < 0$. Let L' be the injective norm of $\nabla^3 f(x)$, then

$$f(x+\epsilon u) \le f(x) - \frac{c\epsilon^2}{2} + \frac{\epsilon^3 L'}{6} + \frac{\epsilon^4 L}{24}.$$

Therefore whenever $\epsilon < \min\{\sqrt{3c/L}, 3c/4L'\}$ we have $f(x+\epsilon u) \le f(x) - \frac{c\epsilon^2}{4}$. The point x cannot be a third order local minimum.

The third case is if $\nabla f(x) = 0$, $\nabla^2 f(x)$ is positive semidefinite, but there is a direction $\|u\| = 1$ such that $u^\top(\nabla^2 f(x))u = 0$, but $[\nabla^3 f(x)](u,u,u) \neq 0$. Without loss of generality we assume $[\nabla^3 f(x)](u,u,u) = c > 0$ (if it is negative we take $-u$), then

$$f(x+\epsilon u) \le f(x) - c\epsilon^3/6 + L\epsilon^4/24.$$

Therefore whenever $\epsilon < 2c/L$ we have $f(x+\epsilon u) \le f(x) - c\epsilon^3/12$ so x cannot be a third order optimal. ■

B.2. Algorithm for Competitive Subspace, Proof of Theorem 16

Theorem 29 (Theorem 16 restated) *There is a universal constant B such that the expected number of iterations of Algorithm 3 is at most 2, and the output of Approx is a unit vector u that satisfies $T(u,u,u) \ge \|Proj_{\mathcal{S}}T\|_F/Q$ for $Q = Bn^{1.5}$.*

Algorithm 4 Algorithm for computing the competitive subspace

Require: Function f, point z, Hessian $M = \nabla^2 f(z)$, third order derivative $T = \nabla^3 f(z)$, approximation ratio Q, Lipschitz Bound L,
Ensure: Competitive subpace $\mathcal{S}(z)$ and $C_Q(z)$.
  Compute the eigendecomposition $M = \sum_{i=1}^{n} \lambda_i v_i v_i^\top$.
  for $i = 1$ **to** n **do**
    Let $\mathcal{S} = \text{span}\{v_i, v_{i+1}, ..., v_n\}$.
    Let $C_Q = \|\text{Proj}_{\mathcal{S}} T\|_F$.
    if $\frac{C_Q^2}{12LQ^2} \geq \lambda_i$ **then**
      return $\mathcal{S}, C_Q$.
    end if
  end for
  return $\mathcal{S} = \emptyset, C_Q = 0$.

Proof We use the anti-concentration property for Gaussian random variables

Theorem 30 (anti-concentration(Carbery and Wright, 2001)) *Let $x \in \mathbb{R}^n$ be a Gaussian variable $x \sim N(0, I)$, for any polynomial $p(x)$ of degree d, there exists a constant κ such that*

$$\Pr[|p(x)| \leq \epsilon \sqrt{\text{Var}[p(x)]}] \leq \kappa \epsilon^{1/d}.$$

In our case $d = 3$ and we can choose some universal constant ϵ such that the probability of $p(x)$ being small is bounded by $1/3$. It is easy to check that the variance is lowerbounded by the Frobenius norm squared, so

$$\Pr[|T(\hat{u}, \hat{u}, \hat{u})| \geq \epsilon \|\text{Proj}_{\mathcal{S}} T\|_F] \geq 2/3.$$

On the other hand with high probability we know the norm of the Gaussian $\hat{u}$ is at most $2\sqrt{n}$. Therefore with probability at least $1/2$, $|T(\hat{u}, \hat{u}, \hat{u})| \geq \epsilon \|\text{Proj}_{\mathcal{S}} T\|_F$ and $\|\hat{u}\| \leq 2\sqrt{n}$, therefore $|T(u, u, u)| \geq \frac{\epsilon}{8n^{1.5}} \|\text{Proj}_{\mathcal{S}} T\|_F$. Choosing $B = 8/\epsilon$ implies the theorem. ■

B.3. Proof of Theorem 24

Theorem 31 (Theorem 24 restated) *It is NP-hard to find a fourth order local minimum of a function $f(x)$, even if f is guaranteed to be well-behaved.*

Proof We reduce the problem of verifying nonnegativenss of degree 4 polynomial to the problem of finding fourth order local minimum.

Given a degree 4 homogeneous polynomial $f(x)$, we can write it as a symmetric fourth order tensor $T \in \mathbb{R}^{n^4}$. Without loss of generality we can rescale T so that $\|T\|_F \leq 1$ and therefore $\|T\| \leq 1$.

Now we define the function $g(x) = f(x) + \|x\|^6$. We first show that this function is well-behaved.

Claim 3 *$g(x)$ is well-behaved.*

Proof Since $g(x)$ is a polynomial with bounded coefficients, clearly it is infinite order differentiable and satisfies condition 2. For condition 1, notice that $g(x) = 0$ and for all $\|x\|_2 > 1$, we have $g(x) \geq \|x\|^6 - \|x\|^4 > 0$ so the global minimizer must be at a point within the unit ℓ_2 ball. Finally, for any $\|x\| = 1$, we know $g(tx) = f(x)t^4 + t^6$ which is always increasing when $t \geq 1$ since $|f(x)| \leq 1$. ■

Next we show if $f(x)$ is nonnegative, then $\mathbf{0}$ is the unique fourth order local minimizer.

Claim 4 *If $f(x)$ is nonnegative, then $\mathbf{0}$ is the unique fourth order local minimizer of $g(x)$.*

Proof Suppose $x \neq 0$ is a local minimizer of $g(x)$ of order at least 1. Let $u = x/\|x\|$. We consider the function $g(tu) = f(u)t^4 + t^6$. Clearly the only first order local minimizer of $g(tu)$ is at $t = 0$. Therefore x cannot be a first order local minimizer of $g(x)$. ■

Finally, we show if $f(x)$ has a negative direction, then all the local minimizer of $g(x)$ must have negative value in f.

Claim 5 *If $f(x)$ is negative for some x, then if x is a fourth order local minimum of $g(x)$ then $f(x) < 0$.*

Proof Suppose $x \neq 0$ is a fourth order local minimum of $g(x)$. Then at least $t = 1$ should be a fourth order local minimum of $g(tx) = f(x)t^4 + t^6\|x\|^6$. This is only possible if $f(x) < 0$.

On the other hand, for $x = 0$, suppose $\|z\| = 1$ is a direction where $f(z) < 0$, then $f(x) - f(x + tz) = f(z)t^4 - t^6 = \Omega(t^4)$, so $x = 0$ is not a fourth order local minimum. ■

The theorem follows immediately from the three claims. ■