DIFFUSION LEARNING IN NON-CONVEX ENVIRONMENTS

Stefan Vlaski[⋆†] *and Ali H. Sayed*[†]

[⋆]Department of Electrical Engineering, University of California, Los Angeles
[†]École Polytechnique Fédérale de Lausanne

ABSTRACT

Driven by the need to solve increasingly complex optimization problems in signal processing and machine learning, recent years have seen rising interest in the behavior of gradient-descent based algorithms in non-convex environments. Most of the works on distributed non-convex optimization focus on the deterministic setting, where exact gradients are available at each agent. In this work, we consider stochastic cost functions, where exact gradients are replaced by stochastic approximations and the resulting gradient noise persistently seeps into the dynamics of the algorithm. We establish that the diffusion algorithm continues to yield meaningful estimates in these more challenging, non-convex environments, in the sense that (a) despite the distributed implementation, restricted to local interactions, individual agents cluster in a small region around a common and well-defined vector, which will carry the interpretation of a network centroid, and (b) the network centroid inherits many properties of the centralized, stochastic gradient descent recursion, including the return of an $O(\mu)$-mean-square-stationary point in at most $O\left(1/\mu^2\right)$ iterations.

Index Terms— Stochastic optimization, adaptation, non-convex, gradient noise, stationary points.

1. INTRODUCTION

The broad objective of distributed adaptation and learning is the solution of global, stochastic optimization problems over distributed networks through localized interactions and in the absence of information about the statistical properties of the data. When constant, rather than diminishing, step-sizes are employed, the resulting algorithms are adaptive in nature and able to adapt to drifts in the problem or data statistics. In this work, we consider a collection of N agents, where each agent is equipped with a stochastic risk of the form $J_k(w) \triangleq \mathbb{E}_x\, Q_k(w;\boldsymbol{x})$ and the objective of the network is to seek the Pareto solution:

$$\min_w J(w), \qquad \text{where } J(w) \triangleq \sum_{k=1}^{N} p_k J_k(w) \tag{1}$$

and p_k are positive weights that are normalized to add up to one and will be specified further below. In (1), the variable $\boldsymbol{x}$ represents data and expectations are computed relative to the distribution of this data. Algorithms for the solution of (1) have been studied extensively over recent years both with inexact [1–3] and exact [4–6] gradients. Here, we shall focus on the following Adapt-then-Combine variation of the diffusion algorithm [2]:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{2a}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^{N} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \tag{2b}$$

where $\widehat{\nabla J}_k(\cdot)$ denotes a stochastic approximation to the true local gradient $\nabla J_k(\cdot)$ and $a_{\ell k}$ are convex combination weights satisfying:

$$a_{\ell k} \geq 0, \quad \sum_{\ell\in\mathcal{N}_k} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{3}$$

where $\mathcal{N}_k$ denotes the set of neighbors of agent k.

Assumption 1 (Strongly-connected graph). *We shall assume that the graph described by the weighted combination matrix $A = [a_{\ell k}]$ is strongly-connected.* □

It then follows from the Perron-Frobenius theorem [2, 7, 8], that A has a single eigenvalue at one while all other eigenvalues are inside the unit circle, so that $\rho(A) = 1$. Moreover, if we let p denote the right-eigenvector of A that is associated with the eigenvalue at one, and if we normalize the entries of p to add up to one, then it also holds that all entries of p are strictly positive, i.e.,

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}}p = 1, \quad p_k > 0 \tag{4}$$

where the $\{p_k\}$ denote the individual entries of the Perron vector, p.

The performance of the diffusion algorithm (2a)–(2b) has been studied extensively in differentiable settings [3, 9], with extensions to multi-task [10], constrained [11] and non-differentiable [12, 13] environments. A common assumption in these works, along with others studying the behavior of distributed optimization algorithms, is that of convexity (or strong-convexity) of the aggregate risk $J(w)$. While many problems of interest such as least-squares estimation [2], logistic regression [2], and support vector machines [14] are convex, there has been increased interest in the optimization of non-convex cost functions. Such problems appear frequently in the design of robust estimators [15] and the training of more complex machine learning architectures such as those involving dictionary learning [16] and artificial neural networks [17].

As a result, recent works have pursued the development and study of optimization algorithms for non-convex problems, both in the deterministic and stochastic setting [18,19]. Works on distributed non-convex optimization so far have largely focused on the *deterministic* setting, where exact gradients are available [20–22]. Other related works consider stochastic gradient approximations, but require a central master node [23], resulting in parallel, rather than a truly distributed implementation, or require a diminishing step-size [24], rendering them inapplicable in scenarios that call for continuous adaptation. In contrast, we shall focus on implementations

This work was supported in part by NSF grant CCF-1524250.
Emails:{stefan.vlaski, ali.sayed}@epfl.ch.

that employ *stochastic* gradient approximations and *constant* step-sizes, and do not require a central communication hub. This is motivated by two considerations. First, computation of the exact gradients $\nabla J_k(\cdot)$ may be infeasible in practice because (a) data may be streaming in, making it impossible to compute $\nabla \mathbb{E}_x\, Q_k(\cdot;\boldsymbol{x})$ in the absence of knowledge about the distribution of the data or (b) the data set, while available as a batch, may be so large that efficient computation of the full gradient is infeasible. Second, and perhaps surprisingly, there have been observations in the literature commenting on the benefit of introducing gradient perturbations into the optimization of non-convex functions [18, 19]. These observations are mainly motivated by the fact that gradient perturbations move iterates "out" of unstable saddle-points and towards more stable local minima, which are generally preferable [17]. Some works have pursued rigorous analytical justifications of this behavior for the class of strict saddle-points [18, 19].

We first establish that in non-convex environments, as is the case in the strongly-convex setting (see for example [3]), the evolution of the iterates $\boldsymbol{w}_{k,i}$ continues to be well-described by the evolution of the weighted centroid vector $\sum_{k=1}^N p_k\,\boldsymbol{w}_{k,i}$, in the sense that the iterates cluster around the network centroid after sufficient iterations. This observation allows us to establish an approximate descent relation for the network centroid, which mirrors that for the centralized stochastic gradient descent up to constants (see for example [18]), and use this relation to establish that the network centroid, around which all agents cluster, will reach an $O(\mu)$-mean-square-stationary point after at most $O\left(1/\mu^2\right)$ iterations.

2. EVOLUTION ANALYSIS

We perform the analysis under the following common assumptions on the gradients and their approximations.

Assumption 2 (Lipschitz gradients). *For each k, the gradient $\nabla J_k(\cdot)$ is Lipschitz, namely, for any $x, y \in \mathbb{R}^M$:*

$$\|\nabla J_k(x) - \nabla J_k(y)\| \leq \delta\|x-y\| \tag{5}$$

In light of (1) *and Jensen's inequality, this implies for the aggregate cost:*

$$\|\nabla J(x) - \nabla J(y)\| \leq \delta\|x-y\| \tag{6}$$

□

Assumption 3 (Bounded gradients). *For each k, the gradient $\nabla J_k(\cdot)$ is bounded, namely, for any $x \in \mathbb{R}^M$:*

$$\|\nabla J_k(x)\| \leq G \tag{7}$$

□

Assumption 4 (Gradient noise process). *For each k, the gradient noise process is defined as*

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) = \widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) - \nabla J_k(\boldsymbol{w}_{k,i-1}) \tag{8}$$

and satisfies

$$\mathbb{E}\,[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})|\boldsymbol{w}_{k,i-1}] = 0 \tag{9a}$$

$$\mathbb{E}\left[\|\,\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^2|\boldsymbol{w}_{k,i-1}\right] \leq \beta^2\|\nabla J_k(\boldsymbol{w}_{k,i-1})\|^2 + \sigma^2 \tag{9b}$$

for some non-negative constants $\{\beta^2, \sigma^2\}$. □

2.1. Centroid recursion

In analyzing the recursions, it turns out to be useful to introduce the following extended quantities, which collect variables from across the network:

$$\boldsymbol{\mathcal{W}}_i \triangleq \mathrm{col}\,\{\boldsymbol{w}_{1,i},\ldots,\boldsymbol{w}_{N,i}\} \tag{10}$$

$$\mathcal{A} \triangleq A \otimes I_M \tag{11}$$

$$\widehat{\boldsymbol{g}}(\boldsymbol{\mathcal{W}}_i) \triangleq \mathrm{col}\left\{\widehat{\nabla J}_1(\boldsymbol{w}_{1,i}),\ldots,\widehat{\nabla J}_N(\boldsymbol{w}_{N,i})\right\} \tag{12}$$

We can then write the diffusion recursion (2a)–(2b) compactly as

$$\boldsymbol{\mathcal{W}}_i = \mathcal{A}^{\mathsf{T}}\left(\boldsymbol{\mathcal{W}}_{i-1} - \mu\widehat{\boldsymbol{g}}(\boldsymbol{\mathcal{W}}_{i-1})\right) \tag{13}$$

Multiplying both sides of (13) by $\left(p^{\mathsf{T}}\otimes I\right)$ from the left, we obtain in light of (4):

$$\left(p^{\mathsf{T}}\otimes I\right)\boldsymbol{\mathcal{W}}_i = \left(p^{\mathsf{T}}\otimes I\right)\boldsymbol{\mathcal{W}}_{i-1} - \mu\left(p^{\mathsf{T}}\otimes I\right)\widehat{\boldsymbol{g}}(\boldsymbol{\mathcal{W}}_{i-1}) \tag{14}$$

Letting $\boldsymbol{w}_{c,i} = \sum_{k=1}^K p_k\,\boldsymbol{w}_{k,i} = \left(p^{\mathsf{T}}\otimes I\right)\boldsymbol{\mathcal{W}}_i$ and exploiting the block-structure of the gradient term, we find:

$$\boldsymbol{w}_{c,i} = \boldsymbol{w}_{c,i-1} - \mu\sum_{k=1}^N p_k\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{15}$$

Note that $\boldsymbol{w}_{c,i}$ is essentially a convex combination of estimates across the network and can be viewed as a weighted centroid. The recursion for $\boldsymbol{w}_{c,i}$ is reminiscent of a stochastic gradient step associated with the aggregate cost $\sum_{k=1}^N p_k J_k(w)$ with the exact gradients $\nabla J_k(\cdot)$ replaced by stochastic approximations $\widehat{\nabla J}_k(\cdot)$ *and* the stochastic gradients evaluated at $\boldsymbol{w}_{k,i-1}$, rather than $\boldsymbol{w}_{c,i-1}$. In fact, we can write:

$$\boldsymbol{w}_{c,i} = \boldsymbol{w}_{c,i-1} - \mu\sum_{k=1}^N p_k\nabla J_k(\boldsymbol{w}_{c,i-1}) - \mu\boldsymbol{d}_{i-1} - \mu\,\boldsymbol{s}_i \tag{16}$$

where we defined the perturbation terms:

$$\boldsymbol{d}_{i-1} \triangleq \sum_{k=1}^N p_k\left(\nabla J_k(\boldsymbol{w}_{k,i-1}) - \nabla J_k(\boldsymbol{w}_{c,i-1})\right) \tag{17}$$

$$\boldsymbol{s}_i \triangleq \sum_{k=1}^N p_k\left(\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) - \nabla J_k(\boldsymbol{w}_{k,i-1})\right) \tag{18}$$

Observe that $\boldsymbol{d}_{i-1}$ arises from the disagreement within the network, and in particular that if each $\boldsymbol{w}_{k,i-1}$ remains close to the network centroid $\boldsymbol{w}_{c,i-1}$, the perturbation will be small in light of the Lipschitz condition (5) on the gradients. The second perturbation term $\boldsymbol{s}_i$ arises from the noise introduced by stochastic gradient approximations at each agent. The remainder of this work is dedicated to establishing that recursion (16) will continue to exhibit some of the desired properties of (centralized) gradient descent, despite the presence of persistent and coupled perturbation terms.

2.2. Network disagreement

To begin with, we study more closely the evolution of the individual estimates $\boldsymbol{w}_{k,i}$ relative to the network centroid $\boldsymbol{w}_{c,i}$. From (16), we have:

$$\boldsymbol{\mathcal{W}}_{c,i} \triangleq \mathbb{1}\otimes\boldsymbol{w}_{c,i} = \left(\mathbb{1}p^{\mathsf{T}}\otimes I\right)\left(\boldsymbol{\mathcal{W}}_{i-1} - \mu\widehat{\boldsymbol{g}}(\boldsymbol{\mathcal{W}}_{i-1})\right) \tag{19}$$

and hence:

$$\begin{aligned}
&\boldsymbol{\mathcal{W}}_i - \boldsymbol{\mathcal{W}}_{c,i} \\
&= \left(\mathcal{A}^{\mathsf{T}} - \mathbb{1}p^{\mathsf{T}} \otimes I\right)\left(\boldsymbol{\mathcal{W}}_{i-1} - \mu\widehat{\boldsymbol{\mathcal{G}}}(\boldsymbol{\mathcal{W}}_{i-1})\right) \\
&\overset{(a)}{=} \left(\mathcal{A}^{\mathsf{T}} - \mathbb{1}p^{\mathsf{T}} \otimes I\right)\left(I - \mathbb{1}p^{\mathsf{T}} \otimes I\right)\left(\boldsymbol{\mathcal{W}}_{i-1} - \mu\widehat{\boldsymbol{\mathcal{G}}}(\boldsymbol{\mathcal{W}}_{i-1})\right) \\
&= \left(\mathcal{A}^{\mathsf{T}} - \mathbb{1}p^{\mathsf{T}} \otimes I\right)\left(\boldsymbol{\mathcal{W}}_{i-1} - \boldsymbol{\mathcal{W}}_{c,i-1} - \mu\widehat{\boldsymbol{\mathcal{G}}}(\boldsymbol{\mathcal{W}}_{i-1})\right) \qquad (20)
\end{aligned}$$

where (a) follows from the equality:

$$\left(\mathcal{A}^{\mathsf{T}} - \mathbb{1}p^{\mathsf{T}} \otimes I\right)\left(I - \mathbb{1}p^{\mathsf{T}} \otimes I\right) = \mathcal{A}^{\mathsf{T}} - \mathbb{1}p^{\mathsf{T}} \otimes I \qquad (21)$$

The recursive relation for the network disagreement allows for the following result.

Lemma 1 (Network disagreement). *Under assumptions 1–4 and for sufficiently small step-sizes μ, the network disagreement is bounded as:*

$$\mathbb{E}\,\|\boldsymbol{\mathcal{W}}_i - \boldsymbol{\mathcal{W}}_{c,i}\|^2 \leq \mu^2 \frac{\lambda_2^2}{(1-\lambda_2)^2}\left(G^2 + \beta^2 G^2 + \sigma^2\right), \text{ for } i \geq i_o. \qquad (22)$$

where $\lambda_2 = \rho\left(\mathcal{A}^{\mathsf{T}} - \mathbb{1}p^{\mathsf{T}} \otimes I\right) + \tau < 1$.

Proof. Omitted due to space limitations. □

This result allows us to bound the perturbation terms encountered in (16).

Lemma 2 (Perturbation bounds). *Under assumptions 1–4 and for sufficiently small step-sizes μ, the perturbation terms are bounded as:*

$$\mathbb{E}\,\|\boldsymbol{d}_{i-1}\|^2 \leq \mu^2 c_\rho \delta^2 p_{\max} \frac{\lambda_2^2}{(1-\lambda_2)^2}\left(G^2 + \beta^2 G^2 + \sigma^2\right) \qquad (23)$$

$$\mathbb{E}\,\|\boldsymbol{s}_{i-1}\|^2 \leq \beta^2 G^2 + \sigma^2 \qquad (24)$$

after sufficient iterations $i \geq i_0$.

Proof. Omitted due to space limitations. □

2.3. Evolution of the network centroid

Having established that after sufficient iterations, all agents in the network will have contracted around the centroid in a small cluster for small step-sizes, we can now leverage $\boldsymbol{w}_{c,i}$ as a proxy for all $\boldsymbol{w}_{k,i}$. Assumption 2 implies the following bound:

$$\begin{aligned}
J(\boldsymbol{w}_{c,i}) \leq J(\boldsymbol{w}_{c,i-1}) &+ \nabla J(\boldsymbol{w}_{c,i-1})^{\mathsf{T}}\left(\boldsymbol{w}_{c,i} - \boldsymbol{w}_{c,i-1}\right) \\
&+ \frac{\delta}{2}\|\boldsymbol{w}_{c,i} - \boldsymbol{w}_{c,i-1}\|^2 \qquad (25)
\end{aligned}$$

This relation, along with (16) and the results from Lemma 2, allow us to establish the following theorem.

Theorem 1 (Descent relation). *Under assumptions 1–4 and for sufficiently small step-sizes, we have:*

$$\mathbb{E}\left\{J(\boldsymbol{w}_{c,i})\,|\,\boldsymbol{w}_{c,i-1}\right\} \leq J(\boldsymbol{w}_{c,i-1}) - \mu c_1 \|\nabla J(\boldsymbol{w}_{c,i-1})\|^2 + \mu^2 c_2 \qquad (26)$$

where

$$\begin{aligned}
c_1 &\triangleq \frac{1 - 2\mu\delta}{2} = O(1) \\
c_2 &\triangleq \frac{\delta}{2}\left(\beta^2 G^2 + \sigma^2\right) + O(\mu) = O(1)
\end{aligned}$$

Proof. Omitted due to space limitations. □

The descent relation (26) has the same form (apart from constants) as the one obtained for centralized gradient descent (see for example [18]) and is useful because it establishes a sufficient condition under which the diffusion recursion (2a)–(2b) is a descent recursion for the network centroid. Specifically, we will have $\mathbb{E}\left\{J(\boldsymbol{w}_{c,i})\,|\,\boldsymbol{w}_{c,i-1}\right\} \leq J(\boldsymbol{w}_{c,i-1})$ whenever:

$$\|\nabla J(\boldsymbol{w}_{c,i-1})\|^2 \geq \mu \frac{c_2}{c_1} \qquad (27)$$

As a corollary we obtain:

Corollary 1 (Stationary points). *Suppose the aggregate cost is bounded from below, i.e., $J(w) \geq J^o$. Then, the centroid $\boldsymbol{w}_{c,i}$ will reach an $O(\mu)$-mean-square-stationary point in at most $O\left(1/\mu^2\right)$ iterations. Specifically for some time $i^\star$, we have:*

$$\mathbb{E}\,\|\nabla J(\boldsymbol{w}_{c,i^\star})\|^2 \leq 2\mu \frac{c_2}{c_1} \qquad (28)$$

and

$$i^\star \leq \left(\frac{J(w_0) - J^o}{c_2}\right)\frac{1}{\mu^2} \qquad (29)$$

Proof. We prove the result by contradiction. First, assume that there is no time $i^\star$ with $\mathbb{E}\,\|\nabla J(\boldsymbol{w}_{c,i^\star})\|^2 \leq 2\mu\frac{c_2}{c_1}$. Then, for all i:

$$\mathbb{E}\,\|\nabla J(\boldsymbol{w}_{c,i})\|^2 \geq 2\mu \frac{c_2}{c_1} \qquad (30)$$

Iterating (26), we obtain:

$$\begin{aligned}
\mathbb{E}\,J(\boldsymbol{w}_{c,i}) &\leq J(w_0) - \mu c_1 \sum_{k=1}^{i}\left(\mathbb{E}\,\|\nabla J(\boldsymbol{w}_{c,k-1})\|^2 - \mu\frac{c_2}{c_1}\right) \\
&\leq J(w_0) - \mu^2 c_2 i \qquad (31)
\end{aligned}$$

and hence $\lim_{i\to\infty}\mathbb{E}\,J(\boldsymbol{w}_{c,i}) \leq -\infty$ which contradicts $J(w) \geq J^o$ for all w. Hence, we conclude that there is a finite, first moment in time $i^\star$ with $\mathbb{E}\,\|\nabla J(\boldsymbol{w}_{c,i^\star})\|^2 \leq 2\mu\frac{c_2}{c_1}$. Since $i^\star$ is the first such time, we still have for $i < i^\star$:

$$\mathbb{E}\,\|\nabla J(\boldsymbol{w}_c)\|^2 \geq 2\mu \frac{c_2}{c_1} \qquad (32)$$

Iterating (26) up to time $i^\star$, we similarly obtain:

$$J^o \leq \mathbb{E}\,J(\boldsymbol{w}_{c,i^\star}) \leq J(w_0) - \mu^2 c_2 i^\star \qquad (33)$$

Rearranging yields the result. □

We conclude that at some time $i^\star \leq O\left(1/\mu^2\right)$, the network centroid will reach an $O(\mu)$-mean-square-stationary point.

3. APPLICATION: ROBUST REGRESSION

Consider a scenario where each agent k in the network observes streaming realizations $\{\boldsymbol{\gamma}(k,i), \boldsymbol{h}_{k,i}\}$ from the linear model:

$$\boldsymbol{\gamma}(k) = \boldsymbol{h}_k^{\mathsf{T}} w^o + \boldsymbol{v}(k) \qquad (34)$$

where $\boldsymbol{\gamma}(k)$ denote scalar observations and $\boldsymbol{v}(k)$ denotes measurement noise. The most commonly used approach for solving such a

problem in a distributed setting is via least-mean-square error estimation, resulting in the local cost functions:

$$J_k^{\mathrm{LS}}(w) = \mathbb{E}\left\|\boldsymbol{\gamma}(k) - \boldsymbol{h}_k^{\mathsf{T}} w\right\|^2 \tag{35}$$

The resulting problem is convex and has been studied extensively in the literature. While effective under the assumption of Gaussian noise, and similar well-behaved noise conditions, this approach is susceptible to outliers caused by heavy-tailed distributions for $\boldsymbol{v}(k)$ [15]. This is caused by the fact that the quadratic risk penalizes errors proportionally to their squared norm, and as such has a tendency to over-correct to outliers, even if they are very rare. Several alternative robust cost functions have been suggested in the literature. We consider two in particular in order to illustrate the advantages of allowing for non-convex costs in the context of robust estimation, namely the Huber loss and Tukey's biweight loss [15]. For ease of notation, let $\boldsymbol{e}(w) \triangleq \boldsymbol{\gamma}(k) - \boldsymbol{h}_k^{\mathsf{T}} w$. Then:

$$Q_k^{\mathrm{H}}(w;\boldsymbol{x}) = \begin{cases} \frac{1}{2}|\boldsymbol{e}(w)|^2, & \text{for } |\boldsymbol{e}(w)| \le c_H \\ c_H|\boldsymbol{e}(w)| - \frac{1}{2}c_H^2, & \text{for } |\boldsymbol{e}(w)| > c_H. \end{cases} \tag{36}$$

$$Q_k^{\mathrm{B}}(w;\boldsymbol{x}) = \begin{cases} \frac{c_B^2}{6}\left(1 - \left(1 - \frac{|\boldsymbol{e}(w)|^2}{c_B^2}\right)^3\right), & \text{for } |\boldsymbol{e}(w)| \le c_B \\ \frac{c_B^2}{6} & \text{otherwise.} \end{cases} \tag{37}$$

where c_H, c_B are tuning constants. The Huber cost is merely convex (and not strongly-convex), while the Tukey loss is non-convex. As such previous results on the performance of the diffusion algorithm (2a)–(2b) are not applicable. Both of them do however satisfy assumptions 1–4 imposed in this work, and as such the results hold. In particular, since the Huber loss $J_k^{\mathrm{H}}(w)$ has a unique, local minimum, which also happens to be locally strongly-convex, we can conclude that the algorithm will approach the global minimum. The Tukey loss on the other hand, is non-convex, and as such a more challenging problem.

The setting for the simulation results is shown in Figure 1.

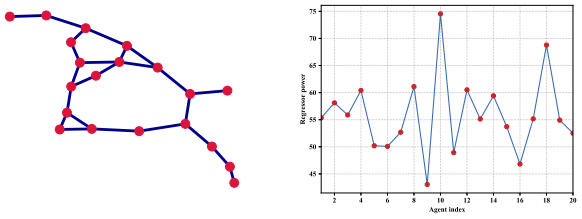

Fig. 1: Graph with $N = 20$ nodes (left) and regressor power $\mathrm{Tr}\,(R_{h,k})$ at each agent (right).

We first show the performance of each strategy in the nominal scenario, where $\boldsymbol{v}(k) \sim \mathcal{N}(0, \sigma_v^2)$. We observe that the distributed strategies outperform the non-cooperative ones, and that despite differences in the rate of convergence, there is negligible difference in the performance of the least-squares, Huber and Tukey variations.

Next, we illustrate the performance in the presence of outliers, modeled as a bimodal distribution with $\boldsymbol{v}(k) \sim (1-\epsilon)\mathcal{N}(0, \sigma_v^2) + \epsilon\mathcal{N}(10, \sigma_v^2)$ and $\epsilon = 0.1$. We observe that the performance of the least-squares solution dramatically deteriorates, as is to be expected in the presence of deviations from the nominal model.

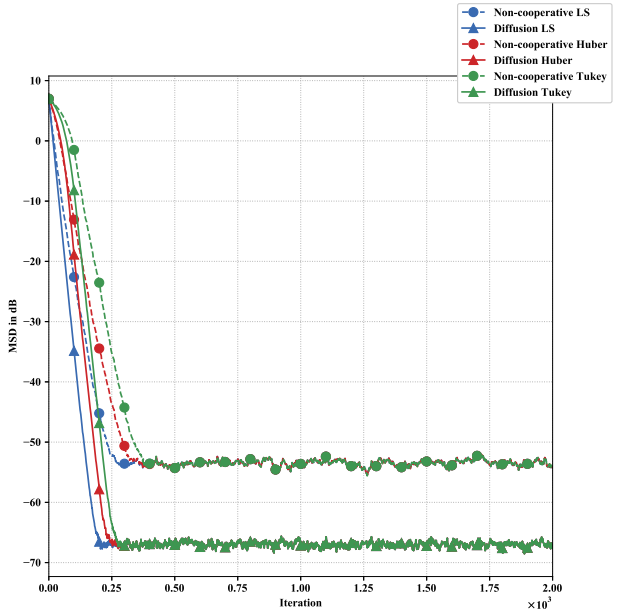

Fig. 2: MSD performance in the absence of outliers.

Fig. 3: MSD performance in the presence of outliers.

4. CONCLUSION

We presented a framework that allows for the study of the behavior of a class of distributed algorithms in non-convex environments and the presence of persistent stochastic perturbations. In particular, we showed that the evolution of the network continues to be well-described by the evolution of the network centroid, in the sense that all iterates $\boldsymbol{w}_{k,i}$ cluster within $O(\mu^2)$ of the network centroid $\boldsymbol{w}_{c,i}$ in the mean-square sense after sufficient iterations. This insight was leveraged to establish a descent relation for the network centroid and to conclude that $\boldsymbol{w}_{c,i}$, and hence $\boldsymbol{w}_{k,i}$ will necessarily reach a point that is $O(\mu)$-mean-square-stationary in at most $O\left(1/\mu^2\right)$ iterations (Corollary 1). Directions for future work include the more careful study of the behavior around saddle-points, and the exploration of further applications, particularly in dictionary learning and deep learning.

5. REFERENCES

[1] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[2] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, July 2014.

[3] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks - Part I: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3487–3517, June 2015.

[4] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[5] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[6] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning – Part II: Convergence analysis," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 724–739, Feb 2019.

[7] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2003.

[8] S. U. Pillai, T. Suel, and S. Cha, "The Perron-Frobenius theorem: Some of its applications," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 62–75, March 2005.

[9] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[10] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6329–6344, Dec 2016.

[11] Z. J. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Trans. on Signal Process.*, vol. 62, no. 15, pp. 3924–3938, Aug. 2014.

[12] S. Vlaski, L. Vandenberghe, and A. H. Sayed, "Diffusion stochastic optimization with non-smooth regularizers," in *Proc. of IEEE ICASSP*, Shanghai, China, March 2016, pp. 4149–4153.

[13] B. Ying and A. H. Sayed, "Performance limits of stochastic sub-gradient learning, Part II: Multi-agent case," *Signal Processing*, vol. 144, pp. 253 – 264, 2018.

[14] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, July 1998.

[15] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*, Cambridge University Press, 2018.

[16] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, March 2011.

[17] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The Loss Surfaces of Multilayer Networks," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, San Diego, May 2015, pp. 192–204.

[18] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—online stochastic gradient for tensor decomposition," in *Proc. of Conference on Learning Theory*, Paris, France, 2015, pp. 797–842.

[19] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in *Proc. of ICML*, Sidney, Australia, Aug. 2017.

[20] P. Di Lorenzo and G. Scutari, "NEXT: in-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, June 2016.

[21] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, Jan. 2019.

[22] Davood Hajinezhad, Mingyi Hong, Tuo Zhao, and Zhaoran Wang, "NESTT: A nonconvex primal-dual splitting method for distributed and stochastic optimization," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, Spain, Dec 2016, pp. 3215–3223.

[23] Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Montreal, Canada, Dec 2015, pp. 2737–2745.

[24] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, Aug. 2017.