Communications in Information and Systems
Volume 20, Number 3, 353–388, 2020

Second-order guarantees in centralized, federated and decentralized nonconvex optimization*

Stefan Vlaski and Ali H. Sayed

In honor of Professor Thomas Kailath on the occasion of his 85th birthday

Rapid advances in data collection and processing capabilities have allowed for the use of increasingly complex models that give rise to nonconvex optimization problems. These formulations, however, can be arbitrarily difficult to solve in general, in the sense that even simply verifying that a given point is a local minimum can be NP-hard [1]. Still, some relatively simple algorithms have been shown to lead to surprisingly good empirical results in many contexts of interest. Perhaps the most prominent example is the success of the backpropagation algorithm for training neural networks. Several recent works have pursued rigorous analytical justification for this phenomenon by studying the structure of the nonconvex optimization problems and establishing that simple algorithms, such as gradient descent and its variations, perform well in converging towards local minima and avoiding saddle-points. A key insight in these analyses is that gradient perturbations play a critical role in allowing local descent algorithms to efficiently distinguish desirable from undesirable stationary points and escape from the latter. In this article, we cover recent results on second-order guarantees for stochastic first-order optimization algorithms in centralized, federated, and decentralized architectures.

1. Learning through optimization

A key desirable feature of automated learning algorithms is the ability to learn models directly from data with minimal need for direct intervention by the designer. This is generally achieved by parameterizing a family of models of sufficient explanatory power through a set of parameters $w \in \mathbb{R}^M$ and subsequently searching for the choice of w^o that fits the data "well", in

*This article provides an overview and summary of results from [2, 3, 4] along with some extensions.

the sense that:

$$w^o \triangleq \arg\min_{w\in\mathbb{R}^M} \mathbb{E}_{\boldsymbol{x}} Q(w;\boldsymbol{x}) \tag{1}$$

In this formulation, the *loss function* $Q(w;\boldsymbol{x})$ denotes a *measure of fit* of the model w for the random data $\boldsymbol{x}$. Hence, the desired model w^o is defined as the one that results in the smallest *expected* risk, where the expectation is taken with respect to the distribution of the data $\boldsymbol{x}$. As we illustrate in a number of examples in the sequel, a vast majority of inference problems fit into the general framework (1).

Example 1 (Loss functions for supervised learning)**.** In the supervised learning setting, the data $\boldsymbol{x} \triangleq \{\boldsymbol{h}, \boldsymbol{\gamma}\}$ can be decomposed into a feature vector $\boldsymbol{h} \in \mathbb{R}^{M_1}$ and a label $\boldsymbol{\gamma}$. When the target variable $\boldsymbol{\gamma} \in \mathbb{R}^{M_2}$ is continuous, this is typically an estimation problem with the objective being to construct an *estimator* $\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h})$ such that the error $\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h}) - \boldsymbol{\gamma}$ is small in some sense with high probability. One popular choice for the loss function $Q(w;\boldsymbol{h},\boldsymbol{\gamma})$ in this case is the squared error loss:

$$Q(w;\boldsymbol{h},\boldsymbol{\gamma}) = \|\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h}) - \boldsymbol{\gamma}\|^2 \tag{2}$$

On the other hand, when $\boldsymbol{\gamma}$ is scalar and discrete as in the binary case $\boldsymbol{\gamma} \in \{-1, 1\}$, the problem becomes a (binary) classification problem, with the objective being to find a classifier $\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h})$ such that with high probability $\text{sign}\{\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h})\} = \boldsymbol{\gamma}$. An example of a popular choice for the loss function in this case is the logistic loss:

$$Q(w;\boldsymbol{h},\boldsymbol{\gamma}) = \log\left(1 + e^{-\boldsymbol{\gamma}\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h})}\right) \tag{3}$$

□

We note that while the choice of the loss function is generally informed by the distribution of the target variable $\boldsymbol{\gamma}$, such as whether it is continuous or discrete, we still need to specify the dependence of $Q(w;\boldsymbol{h},\boldsymbol{\gamma})$ on w. Since in both examples (2) and (3), the loss depends on w through $\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h})$, we can describe this dependence by parameterizing $\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h})$ through w.

Example 2 (Modeling for supervised learning)**.** The most immediate parametrization of $\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h})$ corresponds to the set of linear mappings:

$$\widehat{\boldsymbol{\gamma}}(w;\boldsymbol{h}) \triangleq \boldsymbol{h}^{\mathsf{T}} w \tag{4}$$

Combining the linear model (4) with the quadratic loss (2) results in the minimum mean-square error estimator, while (4) with (3) leads to the logistic regression solution, both of which are convex optimization problems

with efficient solution methods [5]. While convexity of the resulting problem (1) is an appealing property to have, the evident drawback of the linear parametrization (4) is its limited expressive power. Only mappings that correspond to linear combinations of the elements of the feature vector are captured by (4), while non-linear interactions are beyond the scope of this model. For this reason, recent years have seen an increased interest in the utilization of neural networks, which are nested models of the form [6]:

$$\widehat{\gamma}(w;\boldsymbol{h}) \triangleq W_L\sigma\left(W_{L-1}\sigma\left(\ldots\sigma\left(W_1\boldsymbol{h}\right)\right)\right) \tag{5}$$

where the $\left\{W_\ell \in \mathbb{R}^{M_{\ell,1}\times M_{\ell,2}}\right\}$ denote matrices of appropriate dimensions and $\sigma(\cdot)$ denotes an element-wise activation function (usually nonlinear in form). We can collect in w all parameters W_ℓ, i.e., $w \triangleq \operatorname{col}\{\operatorname{vec}\{W_\ell\}\}$ and again recover an instance of (1) for both the quadratic (2) and logistic (3) losses. Models of the form (5), particularly for a suitable size L and dimensions $M_{\ell,1}, M_{\ell,2}$ of hidden layers, are able to model well non-linear classification functions $\widehat{\gamma}(w;\boldsymbol{h})$. However, note that any choice $L \geq 2$ will generally result in a nonconvex loss surface (1). This necessitates the development of performance guarantees of algorithms for algorithms solving (1) under nonconvex environments. □

Example 3 (Unsupervised learning)**.** Not all learning problems present themselves as supervised problems where the objective is to learn a mapping from feature to label. One such example is in the design of recommender systems where users are implicitly clustered and receive recommendations based on the preferences of "similar" other users. A popular approach on this setting revolves around matrix factorization [7]. One such implementation results in:

$$Q(w;\boldsymbol{x}) \triangleq \|\boldsymbol{X} - W_1W_2^{\mathsf{T}}\|^2 + \rho\left(\|W_1\|^2 + \|W_2\|^2\right) \tag{6}$$

where $\boldsymbol{x} \triangleq \operatorname{vec}\{\boldsymbol{X}\}$, $w \triangleq \operatorname{col}\{\operatorname{vec}\{W_1\}, \operatorname{vec}\{W_2\}\}$ and $\rho > 0$ denotes the regularization weights. The matrices W_1, W_2 are generally chosen to be tall, so that $W_1W_2^{\mathsf{T}}$ has low rank, and (6) pursues a low-rank approximation of $\boldsymbol{X}$. □

2. Centralized stochastic optimization

From examples 1–3 we conclude that a large number of learning problems, including linear as well as non-linear regression and classification problems, and unsupervised formulations, can be recovered by specializing the general

stochastic optimization problem (1). The task of designing an effective learning method then boils down to two related decisions: (a) the choice of the learning architecture, which determines the form of the loss $Q(w;\boldsymbol{x})$, and (b) the choice of the optimization strategy, which given realizations of the random variable $\boldsymbol{x}$ yields a high-quality estimate for w^o. For the remainder of this article we will consider the architecture, and hence $Q(w;\boldsymbol{x})$, fixed and will focus on the latter challenge, namely providing performance guarantees for the quality of the estimate of w^o produced by the optimization algorithm for general nonconvex problems. We let:

$$J(w) \triangleq \mathbb{E}_{\boldsymbol{x}} Q(w;\boldsymbol{x}) \tag{7}$$

2.1. Notions of optimality

Loosely speaking, the objective of any (stochastic) optimization algorithm is to produce "high-quality" estimates for the minimizer w^o in (1). When the risk $J(w) \triangleq \mathbb{E}_{\boldsymbol{x}} Q(w;\boldsymbol{x})$ is strongly-convex there is little ambiguity in the quantification of the quality of an estimate, since for strongly-convex costs with constant ν we have [8, Sec. 9.1.2]:

$$\frac{\nu}{2}\|w - w^o\|^2 \leq J(w) - J(w^o) \leq \frac{1}{2\nu}\|\nabla J(w)\|^2 \tag{8}$$

If the risk additionally has δ-Lipschitz gradients, we similarly have [8, Sec. 9.1.2]:

$$\frac{1}{2\delta}\|\nabla J(w)\|^2 \leq J(w) - J(w^o) \leq \frac{\delta}{2}\|w - w^o\|^2 \tag{9}$$

By inspecting these two inequalities we conclude that all three measures of optimality, namely the squared deviation from the minimizer $\|w - w^o\|^2$, the excess risk $J(w) - J(w^o)$, and the squared gradient norm $\|\nabla J(w)\|^2$ are essentially equivalent up to constants that depend on the strong-convexity and Lipschitz parameters δ and ν, respectively. This means that, as long as the problem is reasonably well-conditioned, meaning that the fraction $\frac{\delta}{\nu}$ does not grow too large, the choice of the performance measure is not particularly relevant, since high performance in one measure necessarily implies high performance in both other measures. In other words, any point $w \in \mathbb{R}^M$ with a small gradient norm $\|\nabla J(w)\|^2$, for strongly-convex problems, will essentially be globally optimal in the sense that both the excess risk $J(w) - J(w^o)$ and distance to the minimizer $\|w - w^o\|^2$ will be small.

In the nonconvex setting considered here, and hence in the absence of (8), this is no longer the case as we illustrate in the sequel.

Definition 1 ($O(\mu)$-first-order stationarity)**.** *A point $w \in \mathbb{R}^M$ is $O(\mu)$-first-order stationary if:*

$$\|\nabla J(w)\|^2 \leq O(\mu) \tag{10}$$

These points are technically only approximately *first-order stationary, since exact first-order stationarity would require $\nabla J(w) = 0$. Since we generally refer to $O(\mu)$-first-order stationarity throughout this manuscript, we will drop "approximate" for convenience whenever it is clear from context.* □

In light of relation (9), for costs with δ-Lipschitz gradients, $O(\mu)$-first-order stationarity is a *necessary condition* to ensure $J(w) - J(w^o) \leq O(\mu)$ and $\|w - w^o\|^2 \leq O(\mu)$. However, unless the cost is assumed to additionally be strongly convex, Definition 1 *is not sufficient* to guarantee that the point w has small excess risk $J(w) - J(w^o)$ or small distance to the minimizer $\|w - w^o\|^2$, since establishing sufficiency requires (8) which only holds for *strongly-convex* costs. In fact, the set of $O(\mu)$-first-order stationary points for nonconvex risk functions includes the set of local minima, maxima as well as saddle-points. Nevertheless, many studies of local descent algorithms in nonconvex environments establish performance guarantees by showing that the limiting points of the algorithm are approximately first-order stationary using variations of Definition 1 in both the single-agent and multi-agent settings [9, 10, 11, 12, 13, 14, 15, 16]. These results are reassuring, as first-order stationarity is a necessary condition for local optimality, and hence any algorithm that does not produce a first-order stationary point will necessarily not produce a point with small excess risk, or small distance to the minimizer. Nevertheless, these results cannot ensure that the limiting first-order stationary point does not correspond to a saddle-point, which have been identified as a bottleneck in many nonconvex problems of interest [17]. This observation, following the works [18, 19, 20] motivates us to consider a stronger notion of optimality.

To formulate it, note that our objective is to converge towards points $w \in \mathbb{R}^M$ that are local minima and hence satisfy:

$$J(w) \leq J(w + \Delta w) \tag{11}$$

for all small $\Delta w \in \mathbb{R}^M$. In other words, we would like to avoid approaching points w where there exists $\Delta w \in \mathbb{R}^M$ such that:

$$J(w) > J(w + \Delta w) \tag{12}$$

By introducing the second-order Taylor expansion around w, we can write:

$$\begin{aligned} J(w) - J(w+\Delta w) &\approx -\nabla J(w)^{\mathsf{T}}\Delta w - \Delta w^{\mathsf{T}}\nabla^2 J(w)\Delta w \\ &\approx -\Delta w^{\mathsf{T}}\nabla^2 J(w)\Delta w \end{aligned} \tag{13}$$

where we dropped the linear term $\nabla J(w)^{\mathsf{T}}\Delta w$ since, at first-order stationary points, $\nabla J(w) \approx 0$. Hence, we shall say that w is *second-order* locally optimal according to its *second-order Taylor expansion* if, and only if,

$$\Delta w^{\mathsf{T}}\nabla^2 J(w)\Delta w \geq 0 \tag{14}$$

This requirement is equivalent to $\lambda_{\min}\left(\nabla^2 J(w)\right) \geq 0$. We emphasize that w is *second-order* locally optimal, since expression (13) is only an approximation of $J(w)-J(w+\Delta w)$ based on derivatives up to second-order. Therefore, approaching points w where $\lambda_{\min}(\nabla^2 J(w)) \geq 0$ is desirable. Another way to see this is to note that we also have from (13):

$$\begin{aligned} J(w) &\approx J(w+\Delta w) - \Delta w^{\mathsf{T}}\nabla^2 J(w)\Delta w \\ &\leq J(w+\Delta w) - \lambda_{\min}\left(\nabla^2 J(w)\right)\|\Delta w\|^2 \end{aligned} \tag{15}$$

where equality holds whenever Δw is the eigenvector of $\nabla^2 J(w)$ corresponding to $\lambda_{\min}\left(\nabla^2 J(w)\right)$, i.e., $\nabla^2 J(w)\Delta w = \lambda_{\min}\left(\nabla^2 J(w)\right)\Delta w$. It follows that whenever $\lambda_{\min}\left(\nabla^2 J(w)\right)$ is negative, the larger its magnitude is, the less locally optimal w is. In other words, points w with significantly negative $\lambda_{\min}\left(\nabla^2 J(w)\right)$ are highly undesirable limiting points of a local descent algorithm. Motivated by this discussion, we define the set of τ-second-order stationary points.

Definition 2 (τ-second-order stationarity)**.** *A point $w \in \mathbb{R}^M$ is τ-second-order stationary if it is $O(\mu)$-first-order stationary following Definition 1 and additionally, for some $\tau > 0$,*

$$\lambda_{\min}\left\{\nabla^2 J(w)\right\} > -\tau \tag{16}$$

where $\lambda_{\min}\left\{\nabla^2 J(w)\right\}$ denotes the smallest eigenvalue of the Hessian matrix $\nabla^2 J(w)$. □

We will be focusing on the case when τ is small. Intuitively, points w that satisfy condition (16) are either local minima (e.g., when all eigenvalues of the Hessian matrix are positive) or they are weak saddle-points that are close to local minima (when the smallest eigenvalue is negative but only

by a small amount). Returning to (15), we find that every τ-second-order stationary point w satisfies:

$$J(w) \leq J(w + \Delta w) + \tau \|\Delta w\|^2 \tag{17}$$

Note that, as $\tau \to 0$, the definition of τ-second-order stationarity corresponds to the definition of local optimality (11). The freedom to set any $\tau > 0$, rather than requiring $\tau \to 0$, allows us to set an expectation of local optimality in the sense of (17). This quantity does not appear as a parameter of any of the algorithms presented in this work, but does appear in the expressions on the convergence time (Theorems 2 and 6) as $O(1/\tau)$ meaning that a higher expectation of local optimality requires longer running time of the algorithms, which conforms with intuition. We conclude that, while for non-zero τ not all τ-second-order stationary points are locally optimal, any τ-second-order stationary is *almost* locally optimal for small τ in the sense of (17).

The set of second-order stationary points in Definition 2 is a subset of the set of first-order stationary points in Definition 1. Every second-order stationary point is also first-order stationary, but the additional restriction (16) allows for the exclusion of certain, undesirable, stationary points that do not satisfy (17), such as local maxima and saddle-points. Specifically, by choosing τ small enough, we are able to exclude any first-order stationary point where the smallest eigenvalue of the Hessian is negative and bounded away from zero. These points, which correspond to the complement of Definition 2, are frequently referred to as *strict* saddle-points in the literature due to the requirement for the smallest eigenvalue to be *strictly* negative.

Definition 3 (τ-strict saddle-points)**.** *A point $w \in \mathbb{R}^M$ is a τ-strict saddle-point if it is $O(\mu)$-first-order stationary following Definition 1 and additionally:*

$$\lambda_{\min}\left\{\nabla^2 J(w)\right\} \leq -\tau \tag{18}$$

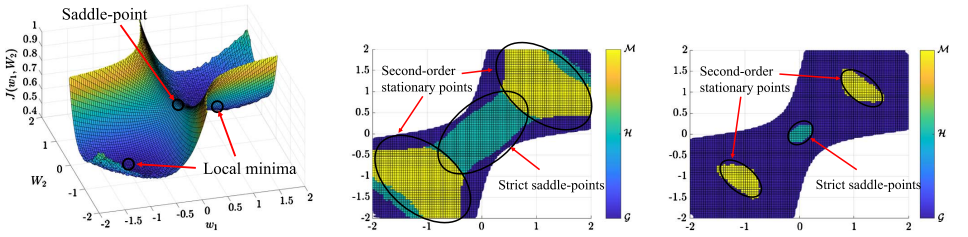
Note that the only difference to Definition 2 is the reversal of inequality (16) to (18). As such, the set of τ-strict saddle-points is precisely the complement of the set of τ-second-order stationary points in the set of first-order stationary points. □

Note that, depending on the choice of the parameter τ, not all saddle-points of the cost $J(w)$ need to be τ-strict saddle-points. If $J(w)$ happens to have a saddle-point with $-\tau \leq \lambda_{\min}\left(\nabla^2 J(w)\right) < 0$, then this particular

saddle-point would not be τ-strict, and in fact would fall under Definition 2 of a τ-second-order stationary point. Nevertheless, so long as τ is small, such saddle-points can intuitively be viewed as "weak" saddle-points, in the sense that they are *almost* locally optimal according to (17).

Under this formal definition, the set of strict saddle-points includes local maxima as well. In fact, if *all* eigenvalues of $\nabla^2 J(w)$ at a first-order stationary point w were bounded from above by $-\tau$, then w would be a *local maximum*. The set of strict saddle-points, however, is larger than the set of local maxima, since *only one* eigenvalue of the Hessian at strict saddle-points is required to be bounded from above by $-\tau$, while other eigenvalues are unrestricted. Hence, the incorporation of second-order information in the definition of stationarity allows us to distinguish between τ-second-order stationary points and τ-strict saddle-points and allows for the exclusion of points with significant local descent direction from the set of potentially optimal points. Furthermore, for many loss functions commonly found in machine learning, such as tensor decomposition [19], matrix completion [21], low-rank recovery [22] and some deep learning formulations [23], *all* saddle-points and local maxima have been shown to have a significant negative eigenvalue in the Hessian, and can hence be excluded from the set of second-order stationary points for sufficiently small, but finite, τ. For such risk functions, *all* τ-second-order stationary points for some small, but finite, τ correspond to local, or even global, minima.

This observation has motivated a number of works to pursue higher-order stationarity guarantees of local descent algorithms by means of second-order information [18, 24, 25], intermediate searches for the negative curvature direction [26, 27, 28], perturbations in the initialization [20, 29, 30] or to the update direction [31, 19, 32, 33, 34, 35, 36, 2, 3, 4], both in the centralized and decentralized setting. Our focus in this manuscript will be on strategies that exploit the presence of perturbations in the update direction to escape from saddle-points. The motivation for this is two-fold. First, in large-scale and online learning problems, the evaluation of exact descent directions is generally infeasible, making the utilization of stochastic gradients, and hence the introduction of stochastic perturbations a necessity. Second, as we shall see, perturbations to the gradient direction can be shown to be sufficient to guarantee *efficient* escape from saddle-points, meaning that the escape-time can be bounded by quantities that scale favorably with problem dimensions and parameters, resulting in simple, yet effective solutions for escaping saddle-points and guaranteeing second-order stationarity without the need to significantly alter the operation of the algorithm.



(a) Sample risk surface. (b) 0.1-stationary points. (c) 0.01-stationary points.

Figure 1: A visual representation of the space decomposition introduced in Definitions 1 through 3 on a sample risk surface with two local minimizers and one saddle-point. The risk surface is depicted in Figure (a). Points in space colored teal $\mathcal{H}$ and yellow $\mathcal{M}$ in Figures (b) and (c) are all 0.1 and 0.01-first-order stationary respectively according to Definition 1. As such, first-order convergence guarantees can only guarantee that the algorithm does not return points in the complement of $\mathcal{G}$, marked in purple, where the norm of the gradient is large. In contrast, we further decompose the set of first-order stationary points into the set of strict saddle-points (set $\mathcal{H}$ in teal) and second-order stationary points (set $\mathcal{M}$, yellow), establish descent for points in $\mathcal{H}$ (teal), and conclude return of second-order stationary points in $\mathcal{M}$ (yellow). Reduction of the step-size parameter μ results in contraction of the set of approximate second-order stationary points around the true local minimizers as is observed from Figure (b) to Figure (c).

2.2. Stochastic gradient descent

One popular first-order approach to pursuing a minimizer for problem (1) can be obtained means of gradient descent, resulting in the recursion:

$$w_i = w_{i-1} - \mu \nabla J(w_{i-1}) \tag{19}$$

The limitation of this recursion lies in the fact that evaluation of the exact gradient of $J(w_{i-1})$ requires statistical information about the random variable $\boldsymbol{x}$ in light of:

$$\nabla J(w_{i-1}) \triangleq \nabla \left(\mathbb{E}_{\boldsymbol{x}} Q(w; \boldsymbol{x}) \right) \tag{20}$$

The most common remedy for this challenge is to instead employ a stochastic approximations of the gradient $\nabla J(w_{i-1})$ based on realizations of the random variable $\boldsymbol{x}$ available at time i. We denote a general stochastic gradient

approximation by $\widehat{\nabla J}(\cdot)$ and iterate:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu\widehat{\nabla J}(\boldsymbol{w}_{i-1}) \tag{21}$$

Observe that we now denote $\boldsymbol{w}_i$ in bold font to emphasize the fact that, by utilizing a stochastic approximation $\widehat{\nabla J}(\cdot)$ based on *realizations* of the random variable $\boldsymbol{x}$ in place of the true gradient $\nabla J(\cdot)$ based on the *distribution* of $\boldsymbol{x}$, the resulting iterates $\boldsymbol{w}_i$ will become stochastic themselves. We will leave the actual specification of the approximation $\widehat{\nabla J}(\boldsymbol{w}_{i-1})$ for the examples and describe performance guarantees under general approximations satisfying fairly general modeling conditions.

2.3. Modeling conditions

We begin by introducing smoothness conditions on both the gradient and Hessian of the risk $J(\cdot)$.

Assumption 1 (Lipschitz gradients)**.** *The gradient $\nabla J(\cdot)$ is Lipschitz, namely, there exists $\delta > 0$ such that for any x, y:*

$$\|\nabla J(x) - \nabla J(y)\| \leq \delta\|x - y\| \tag{22}$$ □

Assumption 2 (Lipschitz Hessians)**.** *The risk $J(\cdot)$ is twice-differentiable and there exists $\rho \geq 0$ such that:*

$$\|\nabla^2 J(x) - \nabla^2 J(y)\| \leq \rho\|x - y\| \tag{23}$$ □

Condition (22) appears commonly in the study of first-order optimality guarantees of (stochastic) gradient algorithms [9, 10, 5]. The Lipschitz condition on the Hessian matrix is not necessary to establish performance bounds in the (strongly-)convex case or first-order stationarity, but can be used to more accurately quantify deviations around the minimizer in steady-state [5], or to establish the escape from saddle-points [19, 32, 35, 3, 4]. The second set of conditions below establishes bounds on the quality of the stochastic gradient approximation $\widehat{\nabla J}(\cdot)$. We define the stochastic gradient noise process:

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \triangleq \nabla J(\boldsymbol{w}_{i-1}) - \widehat{\nabla J}(\boldsymbol{w}_{i-1}) \tag{24}$$

Assumption 3 (Gradient noise process)**.** *The gradient noise process (24) is unbiased with a relative bound on its fourth-moment:*

$$\mathbb{E}\left\{\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \middle| \boldsymbol{w}_{i-1}\right\} = 0 \tag{25}$$

$$\mathbb{E}\left\{\|\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4|\,\boldsymbol{w}_{i-1}\right\} \le \beta^4\|\nabla J(\boldsymbol{w}_{i-1})\|^4 + \sigma^4 \tag{26}$$

for some non-negative constants β^4, σ^4. □

Relation (25) requires that the gradient approximation $\widehat{\nabla J}(\cdot)$ be unbiased. Condition (26) imposes a bound on the fourth moment of the gradient noise, but allows for this bound to grow with the norm of the gradient $\|\nabla J(\boldsymbol{w}_{i-1})\|^4$. Note that, in light of Jensen's inequality and subadditivity of the square root, condition (26) implies and is slightly stronger than:

$$\mathbb{E}\left\{\|\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2|\,\boldsymbol{w}_{i-1}\right\} \le \beta^2\|\nabla J(\boldsymbol{w}_{i-1})\|^2 + \sigma^2 \tag{27}$$

Condition (27) is sufficient to establish limiting first-order stationarity [10], while the fourth-moment condition (26) will allow us to more carefully analyze the dynamics of (21) around first-order stationary points and establish escape from saddle-points, resulting in second-order guarantees. We also impose conditions on the covariance of the gradient noise.

Assumption 4 (Lipschitz covariances)**.** *The gradient noise process has a Lipschitz covariance matrix, i.e.,*

$$R_s(\boldsymbol{w}_{i-1}) \triangleq \mathbb{E}\left\{\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\boldsymbol{s}_i(\boldsymbol{w}_{i-1})^{\mathsf{T}}|\,\boldsymbol{w}_{i-1}\right\} \tag{28}$$

satisfies

$$\|R_s(x) - R_s(y)\| \le \beta_R\|x-y\|^{\gamma} \tag{29}$$

for all x, y*, some* $\beta_R \ge 0$ *and* $0 < \gamma \le 4$. □

Note from the definition of the gradient noise covariance (28), that the distribution of the gradient noise process is a function of the iterate $\boldsymbol{w}_{i-1}$. This, of course, is natural since the gradient noise is defined in (24) as the difference between the true and the approximate gradient *at the current iterate*. The fact that the perturbations introduced into the stochastic recursion (21) are not necessarily identically distributed over time introduces challenges in the study of their cumulative effect. Thankfully, the gradient noise processes induced by most constructions for $\widehat{\nabla J}(\cdot)$ and losses $Q(\cdot,\cdot)$ of interest have a covariance with a Lipschitz-type property (29). This condition ensures that the covariance $R_s(\boldsymbol{w}_{i-1})$ is sufficiently smooth over localized regions in space, resulting in essentially identically distributed gradient

noise perturbations in the short-term and a tractable analysis. It has also been exploited to derive accurate steady-state performance expressions in the strongly-convex setting [5].

In contrast to Assumption 3, which bounds the perturbations induced by employing stochastic gradient approximations *from above*, we will also be imposing a lower bound on the stochastic gradient noise.

Assumption 5 (Gradient noise in strict saddle-points)**.** *Suppose w is an approximate strict-saddle point following Definition 3. Introduce the eigendecomposition of the Hessian matrix as $\nabla^2 J(w) = V\Lambda V^{\mathsf{T}}$ and partition:*

$$V = \begin{bmatrix} V^{\geq 0} & V^{<0} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \Lambda^{\geq 0} & 0 \\ 0 & \Lambda^{<0} \end{bmatrix} \tag{30}$$

where $\Lambda^{\geq 0} \geq 0$ and $\Lambda^{<0} < 0$. Then, we assume that:

$$\lambda_{\min}\left(\left(V^{<0}\right)^{\mathsf{T}} R_s\left(w\right) V^{<0}\right) \geq \sigma_\ell^2 \tag{31}$$

for some $\sigma_\ell^2 > 0$. □

If we construct a local Taylor approximation around the strict saddle-points w, we have:

$$\begin{aligned} J(w+\Delta w) &\approx J(w) + \nabla J(w)^{\mathsf{T}} \Delta w + \Delta w^{\mathsf{T}} \nabla^2 J(w) \Delta w \\ &\approx J(w) + \Delta w^{\mathsf{T}} \nabla^2 J(w) \Delta w \end{aligned} \tag{32}$$

since at strict saddle-points $\nabla J(w) \approx 0$ and, hence, the linear term vanishes. For every Δw in the range of $V^{<0}$, i.e., $\Delta w \triangleq V^{<0} x$, we then have $\Delta w^{\mathsf{T}} \nabla^2 J(w) \Delta w = x^{\mathsf{T}} \left(V^{<0}\right)^{\mathsf{T}} \nabla^2 J(w) V^{<0} x < 0$ by definition of $V^{<0}$, and hence $J(w + \Delta w) < J(w)$. We conclude that the space spanned by $V^{<0}$ corresponds to the local descent directions around the strict saddle-point w. Hence, condition (31) imposes a lower bound on the gradient noise component in the local descent direction (spanned by $V^{<0}$) in the vicinity of saddle-points. It is a notable deviation from the assumptions typically imposed in the *convex* setting. While Assumptions 1–4 are for example all leveraged in deriving steady-state performance expressions in [5] under an additional strong-convexity condition, Assumption 5 is unique to the study of the behavior of stochastic gradient-type algorithms in the vicinity of saddle-points [35, 3, 4] in nonconvex optimization. It may be particularly surprising since the presence of perturbations in the dynamics of gradient-type algorithms are generally understood to be negative side-effects of the utilization

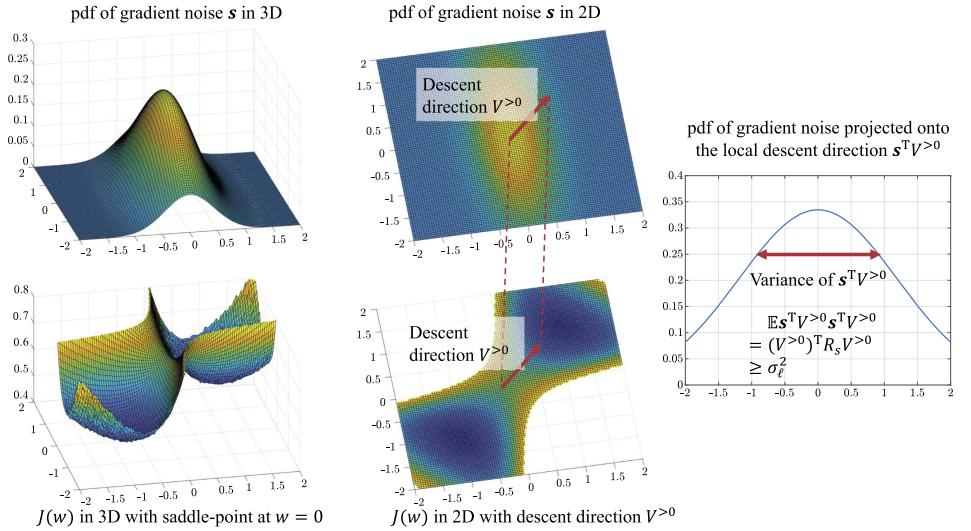


Figure 2: A visual illustration of Assumption 5, which imposes a lower bound on the alignment between gradient noise and the local descent direction. Examples of a probability density function of the gradient noise $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$ (top) and risk function $J(w)$ (bottom) are shown in the left and middle columns, respectively. The risk $J(w)$ exhibits a strict saddle-point at $w = 0$. The local descent direction, which corresponds to $V^{>0}$ in (31) is emphasized as a red arrow in the middle column. Assumption 5 requires some alignment between the local descent direction $V^{>0}$ of the risk (middle bottom) and the probability density function of the gradient noise process (middle top). The quantity $\left(V^{<0}\right)^{\mathsf{T}} R_s\left(w\right) V^{<0}$ in condition (31) in this two-dimensional example corresponds precisely to the variance of the gradient noise after projecting $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$ onto the local descent direction $V^{>0}$, shown in the right column.

of stochastic gradient approximations and result in deterioration of performance, which is generally true for (strongly) convex objectives. When generalizing to nonconvex objectives, as recent analysis has shown [19, 32, 35, 3, 4], the persistent presence of gradient perturbations allows the algorithm to efficiently escape from saddle-points, which are unstable to gradient perturbations, and arrive at local minima, which tend to be more stable to the same types of perturbations. In this sense, condition (31) allows the algorithm to distinguish stable local minima from unstable saddle-points, both of which are first-order stationary points.

As we will see in the examples in the sequel, and the following Section 3, the formulation (21) under the modeling conditions 1–5 is sufficiently general to capture a plethora of first-order stochastic algorithms for the minimization of (7).

Example 4 (Stochastic gradient descent)**.** Suppose we have access to a realization of the data $\boldsymbol{x}_i$ at time i. We can construct a stochastic gradient approximation as:

$$\widehat{\nabla J}^{\mathrm{SGD}}(\boldsymbol{w}_{i-1}) \triangleq \nabla Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_i) \tag{33}$$

Then, condition (25) follows immediately by definition of (33), while (26) can be verified for a number of choices of the loss function $Q(w; \boldsymbol{x})$ and data distributions of $\boldsymbol{x}$. We shall denote the resulting constants:

$$\mathbb{E}\left\{\|\boldsymbol{s}_i^{\mathrm{SGD}}(\boldsymbol{w}_{i-1})\|^4 | \boldsymbol{w}_{i-1}\right\} \leq \beta_{\mathrm{SGD}}^4 \|\nabla J(\boldsymbol{w}_{i-1})\|^4 + \sigma_{\mathrm{SGD}}^4 \tag{34}$$ □

Example 5 (Mini-batch stochastic gradient descent)**.** Suppose we instead have access to a collection of B independent samples $\{\boldsymbol{x}_{b,i}\}_{b=1}^B$ at time i and the computational capacity to compute B gradient operations at every iteration. We can then construct the mini-batch gradient approximation:

$$\widehat{\nabla J}^{\mathrm{B-SGD}}(\boldsymbol{w}_{i-1}) \triangleq \frac{1}{B}\sum_{b=1}^{B} \nabla Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_{b,i}) \tag{35}$$

It again follows that $\widehat{\nabla J}^{\mathrm{B-SGD}}(\boldsymbol{w}_{i-1})$ satisfies (25). For the fourth-order moment can verify by induction over B that:

$$\mathbb{E}\left\{\left\|\boldsymbol{s}_i^{\mathrm{B-SGD}}(\boldsymbol{w}_{i-1})\right\|^4 | \boldsymbol{w}_{i-1}\right\} \leq C_B\left(\frac{\beta_{\mathrm{SGD}}^4}{B^2}\|\nabla J(\boldsymbol{w}_{i-1})\|^4 + \frac{\sigma_{\mathrm{SGD}}^4}{B^2}\right) \tag{36}$$

in terms of the constants β_{SGD}^4 and σ_{SGD}^4 of the single-element stochastic gradient algorithm in example 4, as well as the constant:

$$C_B \triangleq 3 - \frac{2}{B} \leq 3 \tag{37}$$

We observe a B^2-fold decrease in the mean-fourth moment, which implies a B-fold reduction in the second-order moment and complies with our intuition about variance reduction by averaging. For the gradient noise covariance we have:

$$R_s^{\mathrm{B-SGD}}(\boldsymbol{w}_{i-1}) = \frac{1}{B}R_s^{\mathrm{SGD}}(\boldsymbol{w}_{i-1}) \tag{38}$$ □

Example 6 (Perturbed stochastic gradient descent)**.** In the absence of prior knowledge that there is a gradient noise component in the descent direction for every strict saddle-point (Assumption 5), one can always guarantee condition (31) to hold by adding a small perturbation term $\boldsymbol{v}_i$ with positive-definite covariance matrix $R_v \triangleq \mathbb{E}\,\boldsymbol{v}\boldsymbol{v}^{\mathsf{T}} > 0$ as done in [19, 34] to construct:

$$\widehat{\nabla J}^{\mathrm{P-SGD}}(\boldsymbol{w}_{i-1}) \triangleq \widehat{\nabla J}^{\mathrm{SGD}}(\boldsymbol{w}_{i-1}) + \boldsymbol{v}_i = \nabla Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_i) + \boldsymbol{v}_i \tag{39}$$

For the gradient noise covariance we then have:

$$R_s^{\mathrm{P-SGD}}(\boldsymbol{w}_{i-1}) = R_s^{\mathrm{SGD}}(\boldsymbol{w}_{i-1}) + R_v > 0 \tag{40}$$

and hence Assumption 5 is guaranteed to hold. More elaborate constructions, such as only adding an additional perturbation when the iterate $\boldsymbol{w}_{i-1}$ is suspected to be near a first-order stationary point (as done in [32]) are also possible. □

2.4. Second-order guarantee

Due to space limitations, we will only outline the main results that lead to a second-order guarantee for the stochastic approximation algorithm (21) and refer the reader to [2] for a thorough derivation of the result. We begin by formalizing the space decomposition into first and second-order stationary points as well as strict saddle-points.

Definition 4 (Sets)**.** *To simplify the notation in the sequel, we introduce the following sets:*

$$\mathcal{G} \triangleq \left\{ w : \|\nabla J(w)\|^2 \geq \mu \frac{c_2}{c_1} \left(1 + \frac{1}{\pi}\right) \right\} \tag{41}$$

$$\mathcal{G}^C \triangleq \left\{ w : \|\nabla J(w)\|^2 < \mu \frac{c_2}{c_1} \left(1 + \frac{1}{\pi}\right) \right\} \tag{42}$$

$$\mathcal{H} \triangleq \left\{ w : w \in \mathcal{G}^C, \lambda_{\min}\left(\nabla^2 J(w)\right) \leq -\tau \right\} \tag{43}$$

$$\mathcal{M} \triangleq \left\{ w : w \in \mathcal{G}^C, \lambda_{\min}\left(\nabla^2 J(w)\right) > -\tau \right\} \tag{44}$$

where τ is a small positive parameter, c_1 and c_2 are constants:

$$c_1 \triangleq 1 - \mu \frac{\delta}{2}\left(1 + \beta^2\right) = O(1) \tag{45}$$

$$c_2 \triangleq \frac{\delta}{2}\sigma^2 = O(1) \tag{46}$$

and $0 < \pi < 1$ *is a parameter to be chosen. Note that* $\mathcal{G}^C = \mathcal{H} \cup \mathcal{M}$. *For brevity, we also define the probabilities* $\pi_i^{\mathcal{G}} \triangleq \Pr\{\boldsymbol{w}_i \in \mathcal{G}\}$, $\pi_i^{\mathcal{H}} \triangleq \Pr\{\boldsymbol{w}_i \in \mathcal{H}\}$ *and* $\pi_i^{\mathcal{M}} \triangleq \Pr\{\boldsymbol{w}_i \in \mathcal{M}\}$. *Then, for all* i, *we have* $\pi_i^{\mathcal{G}} + \pi_i^{\mathcal{H}} + \pi_i^{\mathcal{M}} = 1$. □

The set $\mathcal{G}^C$ formalizes the set of $O(\mu)$-first-order stationary points in Definition 1 by setting the constant multiplying the step-size μ to $\frac{c_2}{c_1}\left(1 + \frac{1}{\pi}\right)$ where c_1, c_2 are problem-dependent constants. The set $\mathcal{M}$ then corresponds to the set of second-order stationary points in Definition 2 while $\mathcal{H}$ denotes the set of strict saddle-points in Definition 3. For a visualization, we refer the reader back to Fig. 1.

Points in both $\mathcal{G}$ and $\mathcal{H}$ are "undesirable" limiting points in the sense that they have local directions of descent. Our objective is to show that for iterates within both sets, algorithm (21) will continue to descend along the risk (7) by taking local gradient steps. The two sets $\mathcal{G}$ and $\mathcal{H}$ are distinguished by the fact that for points in $\mathcal{G}$, the gradient norm $\|\nabla J(w)\|^2$ is large enough for a single (stochastic) gradient step to be sufficient to guarantee descent in expectation. Points in $\mathcal{H}$ (i.e., strict saddle-points) on the other hand are more challenging since the gradient norm is so small that a single gradient step is no longer sufficient to guarantee descent.

Theorem 1 (Descent in the large-gradient regime [2])**.** *For sufficiently small step-sizes:*

$$\mu \leq \frac{2}{\delta\,(1+\beta^2)} \tag{47}$$

and when the gradient at $\boldsymbol{w}_i$ *is sufficiently large, i.e.,* $\boldsymbol{w}_i \in \mathcal{G}$, *the stochastic gradient recursion* (21) *yields descent in expectation in one iteration, namely,*

$$\mathbb{E}\,\{J(\boldsymbol{w}_{i+1})|\,\boldsymbol{w}_i \in \mathcal{G}\} \leq \mathbb{E}\,\{J(\boldsymbol{w}_i)|\,\boldsymbol{w}_i \in \mathcal{G}\} - \mu^2 \frac{c_2}{\pi} \tag{48}$$

We also establish the following technical result, which bounds the negative effect of the gradient noise close to local minima $w \in \mathcal{M}$*:*

$$\mathbb{E}\,\{J(\boldsymbol{w}_{i+1})|\,\boldsymbol{w}_i \in \mathcal{M}\} \leq \mathbb{E}\,\{J(\boldsymbol{w}_i)|\,\boldsymbol{w}_i \in \mathcal{M}\} + \mu^2 c_2 \tag{49}$$ □

In the vicinity of strict saddle-points $\mathcal{H}$, a more detailed analysis is necessary. Here, it is not the gradient step that ensures descent, but rather the cumulative effect of the gradient noise perturbations to the gradient update. The definition of a strict saddle-point (43) ensures that there is a direction of negative curvature in the local risk surface, while Assumption 5 guarantees

that with some probability the iterate $\boldsymbol{w}_i$ is perturbed towards the descent direction. Together, these conditions allow the algorithm to escape along the descent direction with high probability in a finite number of iterations. This intuition is formalized by constructing a local short-term model based on a local quadratic approximation of the risk surface with identically distributed gradient perturbations and exploiting the smoothness conditions 2 and 4 to bound the approximation error [2, Lemma 3].

Theorem 2 (Descent through strict saddle-points [2])**.** *Beginning at a strict saddle-point $\boldsymbol{w}_i \in \mathcal{H}$ and iterating for i^s iterations after i with*

$$i^s = \frac{\log\left(2M\frac{\sigma^2}{\sigma_\ell^2} + 1 + O(\mu)\right)}{\log(1+2\mu\tau)} \leq O\left(\frac{1}{\mu\tau}\right) \tag{50}$$

guarantees

$$\mathbb{E}\left\{J(\boldsymbol{w}_{i+i^s})\middle|\, \boldsymbol{w}_i \in \mathcal{H}\right\} \leq \mathbb{E}\left\{J(\boldsymbol{w}_i)\middle|\, \boldsymbol{w}_i \in \mathcal{H}\right\} - \frac{\mu}{2}M\sigma^2 + o(\mu) \tag{51}$$

□

Theorem 2 ensures that, even when the norm of the gradient is too small to carry sufficient information about the descent direction, the gradient noise along with the negative local curvature of the risk surface around strict saddle-points is sufficient to guarantee descent in i^s iterations, where the escape-time scales favorably with problem parameters. For example, the escape time scales logarithmically with the problem dimension M, implying that we can expect fast evasion of saddle-points even in high dimensions. Having established descent both in the large-gradient regime and strict-saddle point regime, we can combine the results to conclude eventual second-order stationarity.

Theorem 3 (Second-order guarantee for stochastic gradient descent [2])**.** *Suppose $J(w) \geq J^o$. Then, for sufficiently small step-sizes μ, we have with probability $1-\pi$, that $\boldsymbol{w}_{i^o} \in \mathcal{M}$, i.e., $\|\nabla J(\boldsymbol{w}_{i^o})\|^2 \leq O(\mu)$ and $\lambda_{\min}\left(\nabla^2 J(\boldsymbol{w}_{i^o})\right) \geq -\tau$ in at most i^o iterations, where*

$$i^o \leq \frac{(J(w_0) - J^o)}{\mu^2 c_2} i^s, \tag{52}$$

the quantity $J(w_0) - J^o$ denotes the sub-optimality at the initialization w_0 and i^s denotes the escape time from Theorem 2.

3. Federated learning

In many large-scale applications, data is not available at a central processor, but is instead collected and processed at distributed locations. In this section, we consider a multi-agent setting with a collection of K agents and a central node for parameter aggregation. We associate with each agent its own risk (1), indexed by k:

$$J_k(w) \triangleq \mathbb{E}_{\boldsymbol{x}_k} Q_k(w; \boldsymbol{x}_k) \tag{53}$$

and would like to pursue the minimizer of:

$$J(w) \triangleq \sum_{k=1}^{K} p_k J_k(w) \tag{54}$$

where $p_k > 0$ denote positive weights, normalized to add up to one without loss of generality, i.e., $\sum_{k=1}^{K} p_k = 1$. The problem of distributed minimization of (54) in the presence of a centralized processor, but without the aggregation of raw data $\boldsymbol{x}_{k,i}$ can be achieved using primal and primal-dual approaches [37, 38]. More recently, federated learning has emerged as a framework for the solution of (54) under considerations of asynchrony, heterogeneity, communication and computational restrictions and privacy concerns as they are encountered in practical applications [39]. We show in the sequel that a version of the federated averaging algorithm [39] can be interpreted as the construction of a particular choice for the stochastic gradient approximation $\widehat{\nabla J}(\cdot)$, and hence second-order guarantees can be obtained directly by specializing the results from Section 2. As a baseline, consider the true gradient update to (54), which takes the form:

$$w_i = w_{i-1} - \mu \nabla J(w_{i-1}) = w_{i-1} - \mu \sum_{k=1}^{K} p_k \nabla J_k(w_{i-1}) \tag{55}$$

Just like its single-agent counter-part (19), recursion (55) has the drawback of requiring statistical information about $\boldsymbol{x}_k$ to evaluate the expectations in (53). Additionally, (55) requires full and synchronous participation of all agents k at every iteration by providing (or approximating) the local gradient $\nabla J_k(w_{i-1})$. The former issue can be addressed by employing stochastic gradient approximations based on realizations of data from the distribution of $\boldsymbol{x}_{k,i}$, while the latter issue can be relaxed by allowing for partial participation of agents. To this end, at every iteration i, we sample L agent-indices

without replacement from the set $\{1,\ldots,K\}$ to form $\mathcal{L}$. We introduce the participation indicator function:

$$\mathbb{1}_{k,i} = \begin{cases} 1, & \text{if } k \in \mathcal{L} \text{ at iteration } i, \\ 0, & \text{otherwise.} \end{cases} \tag{56}$$

Then, at every iteration, the global model $\boldsymbol{w}_{i-1}$ is broadcast to participating agents, which collect local data $\{\boldsymbol{x}_{k,i,b}\}_{b=1}^{B}$ and perform the update:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{i-1} - \mu K \mathbb{1}_{k,i} \frac{p_k}{B} \sum_{b=1}^{B} \nabla Q_k(\boldsymbol{w}_{i-1}; \boldsymbol{x}_{k,i,b}) \tag{57}$$

The central processor can then aggregate the intermediate estimates from the participating agents and compute:

$$\boldsymbol{w}_i = \frac{1}{L} \sum_{k=1}^{K} \mathbb{1}_{k,i}\, \boldsymbol{w}_{k,i} \tag{58}$$

Due to the presence of the indicator function $\mathbb{1}_{k,i}$, the aggregation step (58) only requires exchanges with participating agents. Steps (57) and (58) can be combined into:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \frac{K}{L} \sum_{k=1}^{K} \mathbb{1}_{k,i} \frac{p_k}{B} \sum_{b=1}^{B} \nabla Q_k(\boldsymbol{w}_{i-1}; \boldsymbol{x}_{k,b,i}) \tag{59}$$

We argue in the sequel that the approximation:

$$\widehat{\nabla J}(\boldsymbol{w}_{i-1}) \triangleq \frac{K}{L} \sum_{k=1}^{K} \mathbb{1}_{k,i} \frac{p_k}{B} \sum_{b=1}^{B} \nabla Q_k(\boldsymbol{w}_{i-1}; \boldsymbol{x}_{k,b,i}) \tag{60}$$

can be viewed as an instance of the stochastic approximation introduced in Section 2 and hence the results from the single-agent analysis apply. In addition to assuming each $J_k(\cdot)$ satisfies Assumptions 1–5, we will impose the following bound on the agent heterogeneity [36, 3].

Assumption 6 (Bounded gradient disagreement)**.** *For each pair of agents k and ℓ, the gradient disagreement is bounded, namely, for any $x \in \mathbb{R}^M$:*

$$\|\nabla J_k(x) - \nabla J_\ell(x)\| \leq G \tag{61}$$

□

Relation (61) ensures that the disagreement on the local descent direction for any pair of agents is bounded, and is weaker than the more common assumption of uniformly bounded absolute gradients. From Jensen's inequality, we similarly bound the deviation from the aggregate gradient:

$$\begin{aligned}\|\nabla J_k(x) - \nabla J(x)\| &= \left\|\sum_{\ell=1}^{N} p_\ell \left(\nabla J_k(x) - \nabla J_\ell(x)\right)\right\| \\ &\leq \sum_{\ell=1}^{N} p_\ell \left\|\nabla J_k(x) - \nabla J_\ell(x))\right\| \leq G \end{aligned} \tag{62}$$

Example 7 (Federated averaging as a centralized stochastic gradient approximation)**.** We define the local gradient approximation:

$$\widehat{\nabla J}_k(\boldsymbol{w}_{i-1}) \triangleq \frac{K}{L}\frac{\mathbb{1}_{k,i}}{B}\sum_{b=1}^{B}\nabla Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{k,b,i}) \tag{63}$$

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1}) \triangleq \widehat{\nabla J}_k(\boldsymbol{w}_{i-1}) - \nabla J_k(\boldsymbol{w}_{i-1}) \tag{64}$$

We then have:

$$\begin{aligned}\mathbb{E}\left\{\widehat{\nabla J}_k(\boldsymbol{w}_{i-1})\middle|\,\boldsymbol{w}_{i-1}\right\} &\triangleq \frac{K}{L}\frac{\mathbb{E}\{\mathbb{1}_{k,i}\}}{B}\sum_{b=1}^{B}\mathbb{E}\left\{\nabla Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{k,b,i})\middle|\,\boldsymbol{w}_{i-1}\right\} \\ &= \frac{K}{L}\frac{L}{K}\frac{1}{B}\sum_{b=1}^{B}\nabla J_k(\boldsymbol{w}_{i-1}) = \nabla J_k(\boldsymbol{w}_{i-1}) \end{aligned} \tag{65}$$

where we used the fact that $\mathbb{E}\{\mathbb{1}_{k,i}\} = \frac{L}{K}$. For the aggregate risk we then find:

$$\mathbb{E}\left\{\widehat{\nabla J}(\boldsymbol{w}_{i-1})\middle|\,\boldsymbol{w}_{i-1}\right\} = \sum_{k=1}^{K} p_k\,\mathbb{E}\left\{\widehat{\nabla J}_k(\boldsymbol{w}_{i-1})\middle|\,\boldsymbol{w}_{i-1}\right\} \overset{(65)}{=} \nabla J(\boldsymbol{w}_{i-1}) \tag{66}$$

For the fourth-order moment we have:

$$\begin{aligned}\mathbb{E}\left\{\|\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4\middle|\,\boldsymbol{w}_{i-1}\right\} &= \mathbb{E}\left\{\left\|\sum_{k=1}^{K} p_k\,\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\right\|^4 \middle|\,\boldsymbol{w}_{i-1}\right\} \\ &\overset{(a)}{\leq} \sum_{k=1}^{K} p_k\,\mathbb{E}\left\{\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\|^4\middle|\,\boldsymbol{w}_{i-1}\right\} \end{aligned} \tag{67}$$

where (a) follows from the convexity of $\|\cdot\|^4$ and Jensen's inequality. For the local gradient noise terms we have:

$$
\begin{aligned}
&\mathbb{E}\left\{\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\|^4 | \, \boldsymbol{w}_{i-1}\right\} \\
&= \mathbb{E}\left\{\left\|\frac{K}{L}\frac{\mathbb{1}_{k,i}}{B}\sum_{b=1}^{B}\nabla Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{k,b,i}) - \nabla J_k(\boldsymbol{w}_{i-1})\right\|^4 | \, \boldsymbol{w}_{i-1}\right\} \\
&= \mathbb{E}\left\{\left\|\frac{K}{L}\mathbb{1}_{k,i}\left(\frac{1}{B}\sum_{b=1}^{B}\nabla Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{k,b,i}) - \nabla J_k(\boldsymbol{w}_{i-1})\right)\right.\right. \\
&\qquad \left.\left. + \left(\frac{K}{L}\mathbb{1}_{k,i} - 1\right)\nabla J_k(\boldsymbol{w}_{i-1})\right\|^4 | \, \boldsymbol{w}_{i-1}\right\} \\
&\overset{(a)}{\leq} 8\frac{K^4}{L^4}\,\mathbb{E}\left\{\mathbb{1}_{k,i}^4\right\}\cdot\mathbb{E}\left\{\left\|\frac{1}{B}\sum_{b=1}^{B}\nabla Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{k,b,i}) - \nabla J_k(\boldsymbol{w}_{i-1})\right\|^4 | \, \boldsymbol{w}_{i-1}\right\} \\
&\quad + 8\,\mathbb{E}\left\|\frac{K}{L}\mathbb{1}_{k,i} - 1\right\|^4 \|\nabla J_k(\boldsymbol{w}_{i-1})\|^4 \\
&\overset{(b)}{=} 8\frac{K^4}{L^4}\frac{L}{K}\cdot\mathbb{E}\left\{\left\|\frac{1}{B}\sum_{b=1}^{B}\nabla Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{k,b,i}) - \nabla J_k(\boldsymbol{w}_{i-1})\right\|^4 | \, \boldsymbol{w}_{i-1}\right\} \\
&\quad + 8\left(\frac{L}{K}\frac{K-L}{L} + \frac{K-L}{K}\right)\|\nabla J_k(\boldsymbol{w}_{i-1})\|^4 \\
&\overset{(36)}{\leq} \left(8\frac{K^3}{L^3}C_B\frac{\beta_{\mathrm{SGD}}^4}{B^2} + 16\frac{K-L}{K}\right)\|\nabla J_k(\boldsymbol{w}_{i-1})\|^4 + 8\frac{K^3}{L^3}C_B\frac{\sigma_{\mathrm{SGD}}^4}{B^2} \\
&= \left(8\frac{K^3}{L^3}C_B\frac{\beta_{\mathrm{SGD}}^4}{B^2} + 16\frac{K-L}{K}\right)\|\nabla J_k(\boldsymbol{w}_{i-1}) - \nabla J(\boldsymbol{w}_{i-1}) + \nabla J(\boldsymbol{w}_{i-1})\|^4 \\
&\quad + 8\frac{K^3}{L^3}C_B\frac{\sigma_{\mathrm{SGD}}^4}{B^2} \\
&\overset{(c)}{\leq} \left(64\frac{K^3}{L^3}C_B\frac{\beta_{\mathrm{SGD}}^4}{B^2} + 128\frac{K-L}{K}\right)\left(\|\nabla J(\boldsymbol{w}_{i-1})\|^4 + G^4\right) + 8\frac{K^3}{L^3}C_B\frac{\sigma_{\mathrm{SGD}}^4}{B^2} \\
&\qquad\qquad (68) \\
&\overset{(d)}{\leq} \beta_{\mathrm{Fed}}^4\|\nabla J(\boldsymbol{w}_{i-1})\|^4 + \sigma_{\mathrm{Fed}}^4
\end{aligned}
$$

where (a) follows from Jensen's inequality $\|a+b\|^4 \leq 8\|a\|^4 + 8\|b\|^4$, (b) follows from $\Pr\{\mathbb{1}_{k,i} = 1\} = \frac{L}{K}$, (c) again follows from Jensen's inequality

along with Assumption 6 and in (d) we defined:

$$\beta_{\text{Fed}}^4 \triangleq 64\frac{K^3}{L^3}C_B\frac{\beta_{\text{SGD}}^4}{B^2} + 128\frac{K-L}{K} \tag{69}$$

$$\sigma_{\text{Fed}}^4 \triangleq \left(64\frac{K^3}{L^3}C_B\frac{\beta_{\text{SGD}}^4}{B^2} + 128\frac{K-L}{K}\right)G^4 + 8\frac{K^3}{L^3}C_B\frac{\sigma_{\text{SGD}}^4}{B^2} \tag{70}$$

For the aggregate gradient noise term we then have similarly from (67):

$$\mathbb{E}\left\{\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\|^4 | \, \boldsymbol{w}_{i-1}\right\} \leq \beta_{\text{Fed}}^4\|\nabla J(\boldsymbol{w}_{i-1})\|^4 + \sigma_{\text{Fed}}^4 \tag{71}$$

and hence the federated learning algorithm (59) satisfies Assumption 3. □

3.1. Second-order guarantees for federated averaging

Having established in Example 7 that the federated averaging algorithm (59) satisfies Assumption 3 with constants (69)–(70), we can specialize Theorem 3 to recover second-order guarantees for (59).

Corollary 1 (Second-order guarantee for federated averaging)**.** *Suppose $J(w) \geq J^o$. Then, for sufficiently small step-sizes μ, the federated averaging algorithm (59) with probability $1-\pi$ generates a point $\boldsymbol{w}_{i^o} \in \mathcal{M}$ with:*

$$\|\nabla J(\boldsymbol{w}_{i^o})\|^2 \leq O\left(\mu\sigma_{\text{Fed}}^2\right) \tag{72}$$

and $\lambda_{\min}\left(\nabla^2 J(\boldsymbol{w}_{i^o})\right) \geq -\tau$ in at most i^o iterations, where

$$i^o \leq \frac{(J(w_0)-J^o)}{\mu^2 c_2 \pi} i^s, \tag{73}$$

the quantity $J(w_0)-J^o$ describes the initial sub-optimality, and i^s denotes the escape time from Theorem 2:

$$i^s = \frac{\log\left(2M\frac{\sigma_{\text{Fed}}^2}{\sigma_{\text{Fed},\ell}^2} + 1 + O(\mu)\right)}{\log(1+2\mu\tau)} \leq O\left(\frac{1}{\mu\tau}\right) \tag{74}$$

□

The constant σ_{Fed}^2 is the dominant constant determining the level of accuracy guaranteed by the result in Corollary 1. Its expression in (70) quantifies the dependence on the various federated learning parameters such as the participation rate $\frac{L}{K}$ and the local mini-batch sizes B.

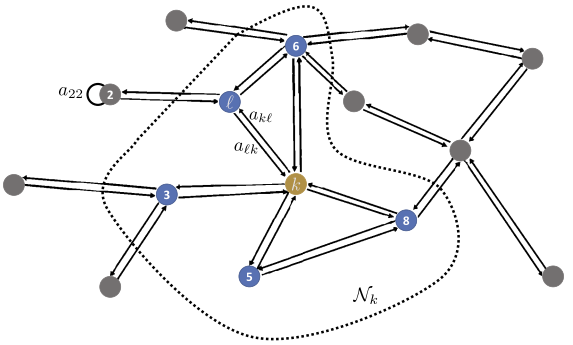


Figure 3: A sample network with an emphasis on the neighborhood $\mathcal{N}_k$ of node k. Node k can aggregate information from only its neighbors in $\mathcal{N}_k$, with $a_{\ell k}$ denoting the weight given by node k to information from ℓ. Double-arrows indicate the asymmetric flow of information, since we allow for $a_{\ell k} \neq a_{k\ell}$.

4. Decentralized learning

While fusion-center based approaches, such as (59), are an effective approach to learning from distributed data sources $\boldsymbol{x}_k$ without the need to exchange raw data, and instead relying solely on the exchange of local models $\boldsymbol{w}_{k,i}$, they have the drawback of nevertheless requiring some form of central aggregation. In this section, we relax this requirement. We continue to consider a collection of K agents, and continue to pursue solutions to (54), repeated here for reference:

$$J(w) \triangleq \sum_{k=1}^{K} p_k J_k(w) \tag{75}$$

In contrast to the federated learning framework, which allows for the central aggregation of (a subset of) intermediate parameter estimates, we now consider the agents to be connected via a graph topology, restricting the flow of information. A sample graph is provided in Fig. 3. It is then natural to ask whether the collection of agents can still pursue a solution of (75) despite being restricted to performing only local computations and *exchanges of information over neighborhoods*. The answer is indeed affirmative, so long as the network linking the agents is connected, allowing for information to diffuse through the entire network through repeated local

aggregations. The solution can then be pursued through a plethora of decentralized strategies, including primal [40, 41, 5, 42, 43, 44, 12, 45, 46] and (primal-)dual [47, 48, 49, 50, 51, 52, 53] frameworks. A detailed discussion of the properties of these strategies, primarily studied in the convex setting, is beyond the scope of this work. We will instead focus on the diffusion strategy [42, 5, 43], and discuss its second-order guarantees in nonconvex environments. The diffusion strategy takes the form:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{76a}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^{N} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \tag{76b}$$

Note that the strategy has an "adapt-then-combine" form where in step (76a), agent k takes a local descent step using the approximation $\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1})$ based on its locally available data to generate an intermediate estimate $\boldsymbol{\phi}_{k,i}$. The adaptation step is followed by a combination step where agent k performs a convex combination of the intermediate estimates $\boldsymbol{\phi}_{\ell,i}$ using the weights $a_{\ell k}$ to form $\boldsymbol{w}_{k,i}$. We shall make the following assumption on the combination weights.

Assumption 7 (Strongly-connected graph)**.** *The graph described by the weighted combination matrix $A = [a_{\ell k}]$ is strongly-connected [5]. This means that there exists a path with nonzero weights between any two agents in the network and, moreover, at least one agent has a nontrivial self-loop, $a_{kk} > 0$. The combination weights satisfy:*

$$a_{\ell k} \geq 0, \quad \sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{77}$$

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

Relation (77) ensures that (76b) indeed carries the interpretation of a convex combination, and can be evaluated by collecting intermediate estimates $\boldsymbol{\phi}_{\ell,i}$ only from nodes in the immediate neighborhood $\ell \in \mathcal{N}_k$. Strong connectivity of the graph on the other hand, in light of the Perron-Frobenius theorem [54, 55, 5], ensures that the combination matrix A has a single eigenvalue at one with all other eigenvalues strictly within the unit circle. The right eigenvalue of A, denoted by p can be normalized so that its elements are strictly positive and add up to one [5]:

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}}p = 1, \quad p_k > 0 \tag{78}$$

where the $\{p_k\}$ denote the individual entries of the Perron vector, p. In the strongly-convex case, it is well known that the diffusion strategy (76a)–(76b) converges to the minimizer of (75) in the mean-square sense where the weights p_k in (75) correspond to the entries of the Perron vector p in (78) of the combination matrix A [42, 5]. It is common to choose A to be symmetric, resulting in $p_k = \frac{1}{K}$ for all k and equal contribution of all nodes in (75). Allowing for more general left- instead of only doubly-stochastic matrices provides the designer with the additional flexibility to, for example, assign larger weight to nodes with higher quality of data, a fact that has been exploited both in the strongly-convex [5] and nonconvex [56] setting. In both cases these strategies exploit that for any connected graph, a combination matrix satisfying Assumption 7 can be designed in a decentralized manner to have an arbitrary p as its Perron vector $Ap = p$ [5, Eq. (8.96)].

5. Network dynamics

The fact that p corresponds to a right eigenvector of the combination matrix A implies for the weighted centroid $\boldsymbol{w}_{c,i} \triangleq \sum_{k=1}^{K} p_k \boldsymbol{w}_{k,i}$ [44, 5]:

$$\boldsymbol{w}_{c,i} = \boldsymbol{w}_{c,i-1} - \mu \sum_{k=1}^{K} p_k \widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{79}$$

Examination of (79) shows that $\boldsymbol{w}_{c,i}$ evolves *almost* according to a (stochastic) gradient recursion relative to the aggregate cost (75) with the subtle difference that the gradient approximations $\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1})$ are evaluated at the local iterates $\boldsymbol{w}_{k,i-1}$ instead of the centroid $\boldsymbol{w}_{c,i-1}$. Nevertheless, as long as the collection of iterates $\boldsymbol{w}_{k,i-1}$ do not deviate too much from each other, and hence from the (weighted) average $\boldsymbol{w}_{c,i} = \sum_{k=1}^{K} p_k \boldsymbol{w}_{k,i}$, one would expect the evolution of (79) to carry similar performance guarantees to the single-agent and federated solutions (21) and (59), respectively. This has been rigorously established in the strongly-convex case [5, 44, 57]. In this work, we present more recent extensions to nonconvex risks and second-order guarantees.

Theorem 4 (Network disagreement (4th order) [3])**.** *Suppose each local $J_k(\cdot)$ and stochastic gradient approximation $\widehat{\nabla J}_k(\cdot)$ satisfy Assumptions 1–6 with $\beta = 0$. Furthermore, assume the combination matrix A satisfies Assumption 7 with Jordan decomposition $A = V_\epsilon J V_\epsilon^{-1}$:*

$$V_\epsilon = \begin{bmatrix} p & V_R \end{bmatrix}, \quad J = \begin{bmatrix} 1 & 0 \\ 0 & J_\epsilon \end{bmatrix}, \quad V_\epsilon^{-1} = \begin{bmatrix} \mathbb{1}^{\mathsf{T}} \\ V_L^{\mathsf{T}} \end{bmatrix} \tag{80}$$

Collect the iterates $\boldsymbol{w}_{k,i}$ *across the network into* $\boldsymbol{\mathcal{W}}_i \triangleq \operatorname{col}\{\boldsymbol{w}_{1,i},\ldots,\boldsymbol{w}_{K,i}\}$. *Then, the network disagreement is bounded after sufficient iterations* $i \geq i_o$ *by:*

(81)
$$\mathbb{E}\left\|\boldsymbol{\mathcal{W}}_i - \left(\mathbb{1}p^{\mathsf{T}}\otimes I\right)\boldsymbol{\mathcal{W}}_i\right\|^4 \leq \mu^4\|\mathcal{V}_L\|^4\frac{\left\|J_\epsilon^{\mathsf{T}}\right\|^4}{\left(1-\|J_\epsilon^{\mathsf{T}}\|\right)^4}\|\mathcal{V}_R^{\mathsf{T}}\|^4 N^2\left(G^4+\sigma^4\right)+o(\mu^4)$$

where $\mathcal{V}_L\otimes I_M$, $\mathcal{V}_R = V_R\otimes I_M$ *and*

$$i_o = \frac{\log\left(o(\mu^4)\right)}{\log\left(\|J_\epsilon^{\mathsf{T}}\|\right)} \tag{82}$$

and $o(\mu^4)$ *denotes a term that is higher in order than* μ^4. □

To develop some intuition about the implications of (81), observe that we can bound:

$$\begin{aligned}\frac{1}{N}\sum_{k=1}^{K}\mathbb{E}\,\|\,\boldsymbol{w}_{k,i}-\boldsymbol{w}_{c,i}\,\|^2 &= \frac{1}{N}\,\mathbb{E}\left\|\boldsymbol{\mathcal{W}}_i-\left(\mathbb{1}p^{\mathsf{T}}\otimes I\right)\boldsymbol{\mathcal{W}}_i\right\|^2\\ &\overset{(a)}{\leq}\frac{1}{N}\sqrt{\mathbb{E}\,\|\boldsymbol{\mathcal{W}}_i-\left(\mathbb{1}p^{\mathsf{T}}\otimes I\right)\boldsymbol{\mathcal{W}}_i\|^4}\\ &\overset{(b)}{\leq}\mu^2\|\mathcal{V}_L\|^2\frac{\left\|J_\epsilon^{\mathsf{T}}\right\|^2}{\left(1-\|J_\epsilon^{\mathsf{T}}\|\right)^2}\|\mathcal{V}_R^{\mathsf{T}}\|^2 N\left(G^2+\sigma^2\right)+o(\mu^2)\end{aligned} \tag{83}$$

where (a) follows from Jensen's inequality along with convexity of $\|\cdot\|^2$ and (b) follows from sub-additivity of the square root $\sqrt{a+b}\leq\sqrt{a}+\sqrt{b}$. We hence conclude that (83) bounds the average deviation of the local iterates $\boldsymbol{w}_{k,i}$ from the centroid $\boldsymbol{w}_{c,i}$ in the mean-square sense by a term that is on the order of μ^2, which is small enough to be negligible for sufficiently small step-sizes μ. This allows us to derive essentially the same performance guarantees for the network centroid $\boldsymbol{w}_{c,i}$ as for the centralized recursion (21), after accounting for the small and controllable deviation (83). We make a minor adjustment to the space decomposition from Definition 4.

Definition 5 (Sets)**.** *We continue with the decomposition into* $\mathcal{G}$, $\mathcal{H}$ *and* $\mathcal{M}$ *from relations* (41)–(44) *in Definition 4 and only adjust expression* (45) *for* c_1 *to:*

$$c_1 \triangleq \frac{1}{2}\left(1-2\mu\delta\right) = O(1) \tag{84}$$

and $0 < \pi < 1$ *is a parameter to be chosen. Note that* $\mathcal{G}^C = \mathcal{H} \cup \mathcal{M}$. *We also define the probabilities* $\pi_i^{\mathcal{G}} \triangleq \Pr\{\boldsymbol{w}_{c,i} \in \mathcal{G}\}$, $\pi_i^{\mathcal{H}} \triangleq \Pr\{\boldsymbol{w}_{c,i} \in \mathcal{H}\}$ *and* $\pi_i^{\mathcal{M}} \triangleq \Pr\{\boldsymbol{w}_{c,i} \in \mathcal{M}\}$. *Then for all* i, *we have* $\pi_i^{\mathcal{G}} + \pi_i^{\mathcal{H}} + \pi_i^{\mathcal{M}} = 1$. □

Note that the only difference between Definitions 4 and 5 is in the definition of c_1 in (45) and (84). This variation is motivated by the technical details of the arguments leading to the descent relations that follow, but ultimately does not change the implications of the result. We then obtain the decentralized versions to the centralized descent Theorems 1 through 3, established in [3, 4].

Theorem 5 (Descent relation [3])**.** *Beginning at* $\boldsymbol{w}_{c,i-1}$ *in the large gradient regime* $\mathcal{G}$, *we can bound:*

$$\begin{aligned} &\mathbb{E}\{J(\boldsymbol{w}_{c,i})|\,\boldsymbol{w}_{c,i-1} \in \mathcal{G}\} \\ &\leq \mathbb{E}\{J(\boldsymbol{w}_{c,i-1})|\,\boldsymbol{w}_{c,i-1} \in \mathcal{G}\} - \mu^2 \frac{c_2}{\pi} + \frac{O(\mu^3)}{\pi_{i-1}^{\mathcal{G}}} \end{aligned} \tag{85}$$

as long as $\pi_{i-1}^{\mathcal{G}} = \Pr\{\boldsymbol{w}_{c,i-1} \in \mathcal{G}\} \neq 0$ *where the relevant constants are listed in Definition 5. On the other hand, beginning at* $\boldsymbol{w}_{c,i-1} \in \mathcal{M}$, *we can bound:*

$$\begin{aligned} &\mathbb{E}\{J(\boldsymbol{w}_{c,i})|\,\boldsymbol{w}_{c,i-1} \in \mathcal{M}\} \\ &\leq \mathbb{E}\{J(\boldsymbol{w}_{c,i-1})|\,\boldsymbol{w}_{c,i-1} \in \mathcal{M}\} + \mu^2 c_2 + \frac{O(\mu^3)}{\pi_{i-1}^{\mathcal{M}}} \end{aligned} \tag{86}$$

as long as $\pi_{i-1}^{\mathcal{M}} = \Pr\{\boldsymbol{w}_{c,i-1} \in \mathcal{M}\} \neq 0$. □

Theorem 6 (Descent through strict saddle-points [4])**.** *Suppose* $\Pr\{\boldsymbol{w}_{c,i} \in \mathcal{H}\} \neq 0$, *i.e.,* $\boldsymbol{w}_{c,i}$ *is approximately stationary with significant negative eigenvalue. Then, iterating for* i^s *iterations after* i *with*

$$i^s = \frac{\log\left(2M\frac{\sigma_a^2}{\sigma_i^2} + 1\right)}{\log(1 + 2\mu\tau)} \leq O\left(\frac{1}{\mu\tau}\right) \tag{87}$$

guarantees

$$\begin{aligned} &\mathbb{E}\{J(\boldsymbol{w}_{c,i+i^s})|\,\boldsymbol{w}_{c,i} \in \mathcal{H}\} \\ &\leq \mathbb{E}\{J(\boldsymbol{w}_{c,i})|\,\boldsymbol{w}_{c,i} \in \mathcal{H}\} - \frac{\mu}{2} M\sigma^2 + o(\mu) + \frac{o(\mu)}{\pi_i^{\mathcal{H}}} \end{aligned} \tag{88}$$

□

Theorem 7 (Second-order guarantee for diffusion [4])**.** *For sufficiently small step-sizes μ, we have with probability $1-\pi$, that $\boldsymbol{w}_{c,i^o} \in \mathcal{M}$, i.e., $\|\nabla J(\boldsymbol{w}_{c,i^o})\|^2 \leq O(\mu)$ and $\lambda_{\min}\left(\nabla^2 J(\boldsymbol{w}_{c,i^o})\right) \geq -\tau$ in at most i^o iterations, where*

$$i^o \leq \frac{(J(w_{c,0}) - J^o)}{\mu^2 c_2} i^s + o(\mu^{-1}) \tag{89}$$

and i^s denotes the escape time from Theorem 6, i.e.,

$$i^s = \frac{\log\left(2M\frac{\sigma^2}{\sigma_i^2}+1\right)}{\log(1+2\mu\tau)} \leq O\left(\frac{1}{\mu\tau}\right) \tag{90}$$

□

Comparing Theorems 5–7 to Theorems 1–3 we note that the descent and second-order stationarity guarantees for the network centroid $\boldsymbol{w}_{c,i}$ generated by the diffusion algorithm (76a)–(76b) are essentially the same as those for the ordinary stochastic gradient descent recursion (21) after adjusting the constants to account for the decentralized nature. Theorem 4, on the other hand, ensures that all iterates $\boldsymbol{w}_{k,i}$ will closely track the network centroid $\boldsymbol{w}_{c,i}$ after sufficient iterations, and hence each agent k in the network with inherit the second-order guarantees of $\boldsymbol{w}_{c,i}$.

6. Simulation example

We illustrate the theoretical results in this work on a simple example, motivated by neural network learning and used as a benchmark in [3, 4, 2]. Given a feature vector $\boldsymbol{h} \in \mathbb{R}^M$ and binary label $\boldsymbol{\gamma} \in \{0,1\}$, we model a learning rule $\widehat{\gamma}(\boldsymbol{h};\cdot)$ through a neural network with one linear hidden layer and a logistic activation function at the output layer, taking the form:

$$\widehat{\gamma}(\boldsymbol{h}; w_1, W_2) \triangleq \frac{1}{1+e^{-w_1^{\mathsf{T}} W_2 \boldsymbol{h}}} \tag{91}$$

where $w_1 \in \mathbb{R}^N, W_2 \in \mathbb{R}^{N\times M}$ denote the model parameters. We employ the cross-entropy loss:

$$Q(w_1, W_2; \boldsymbol{h}, \boldsymbol{\gamma}) = -\boldsymbol{\gamma}\log\left(\widehat{\gamma}(\boldsymbol{h}; w_1, W_2)\right) - (1-\boldsymbol{\gamma})\log\left(1-\widehat{\gamma}(\boldsymbol{h}; w_1, W_2)\right) \tag{92}$$

The risk obtained after taking expectations and adding regularization is:

$$J(w_1, W_2) \triangleq \mathbb{E}\, Q(w_1, W_2; \boldsymbol{h}, \boldsymbol{\gamma}) + \frac{\rho}{2}\|w_1\|^2 + \frac{\rho}{2}\|W_2\|^2 \tag{93}$$

This risk function is nonconvex, and for $M = N = 1$ has two local minima in the positive and negative quadrants, respectively with a single strict saddle-point at $w_1 = W_2 = 0$. This risk surface was used in Figures 1 and 2 to illustrate the modeling conditions for the theorems in this work. We now illustrate the practical performance of the centralized strategy (21), the federated algorithm (59), as well as the decentralized diffusion strategy (76a)–(76b) and verify that the second-order guarantees established in Theorems 3 and 7 indeed hold.

We consider a collection of $K = 50$ agents, each sampling independent pairs $\{\boldsymbol{h}_k, \boldsymbol{\gamma}(k)\}$ once per iteration. Although it is not a requirement of the analysis, for simplicity, we consider in this example a homogeneous data setting, where pairs $\{\boldsymbol{h}_k, \boldsymbol{\gamma}(k)\}$ are identically distributed for all agents and $J_k(w_1, W_2) = J(w_1, W_2)$ for all k. The agents are linked by a random graph with mixing rate $\rho\left(A - \mathbb{1}p^{\mathsf{T}}\right) = 0.956$ with combination weights giving equal weight to all neighbors. We collect $w \triangleq \operatorname{col}\{w_1, \operatorname{vec}\{W_2\}\}$. For the centralized solution (21), the stochastic gradient approximation is constructed by aggregating data from *all* agents:

$$\widehat{\nabla J}(w) \triangleq \sum_{k=1}^{K} p_k \nabla Q(w; \boldsymbol{h}_k, \boldsymbol{\gamma}(k)) + \rho w + \boldsymbol{v} \cdot \operatorname{col}\{1, 1\} \tag{94}$$

where the random perturbation $\boldsymbol{v} \sim \mathcal{N}(0, 1)$ was added to ensure that Assumption 5 holds. For the federated implementation (59), each of the $L = 10$ participating agents at each iteration construct:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_i - \mu K p_k \mathbb{1}_{k,i} \left((\nabla Q(w; \boldsymbol{h}_k, \boldsymbol{\gamma}(k)) + \rho w + \boldsymbol{v} \cdot \operatorname{col}\{1, 1\}) \right) \tag{95}$$

The intermediate estimates $\boldsymbol{w}_{k,i}$ are then fused according to (58). For the decentralized implementation, each agent constructs:

$$\widehat{\nabla J_k}(w) \triangleq \nabla Q(w; \boldsymbol{h}_k, \boldsymbol{\gamma}(k)) + \rho w + \boldsymbol{v} \cdot \operatorname{col}\{1, 1\} \tag{96}$$

and then updates according to (76a)–(76b). All iterates for all three strategies are initialized at $\{0.8, -0.8\}$. As predicted by the theoretical results, all three strategies are able to escape the saddle-point at $w_1 = W_2 = 0$. Detailed performance is shown in Fig. 4.

7. Conclusion

In this manuscript we presented recent results from [2, 3, 4] establishing second-order guarantees for stochastic descent algorithms in centralized,

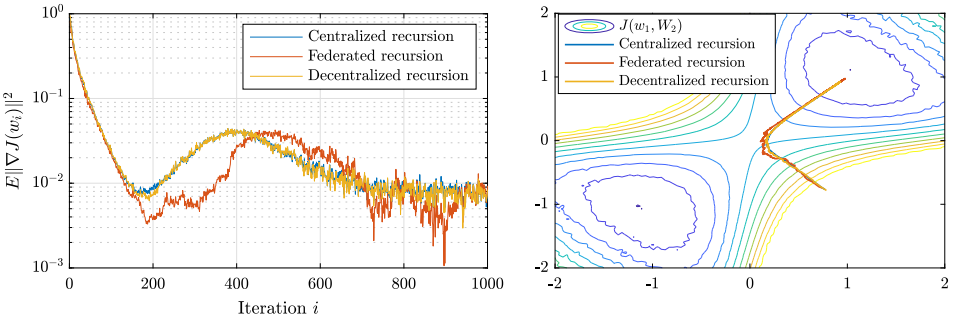


Figure 4: (left) Evolution of the gradient norm. (right) Evolution of the iterates. Around iteration 200, all three algorithms approach the saddle-point, where the norm of the gradient drops and the evolution slows. Due to the presence of gradient perturbations, all three algorithms are able to escape from the saddle-point and eventually reach a local minimum. The decentralized solution (76a)–(76b) closely tracks the centralized algorithm (21), while the federated algorithm (59) exhibits slightly higher variance due to the scaled step-size to account for partial agent participation.

federated, and decentralized settings. Two key conclusions emerge. First, we found that in all cases, simple first-order descent algorithms are able to yield second-order optimal solutions, which exclude saddle-points and correspond to local or even global minima in many problems of interest, so long as their recursions are subjected to sufficient perturbations. These perturbations are critical in ensuring that the recursions do not spend extraordinary amounts of time near saddle-points, where the progress of unperturbed gradient recursions is slow [29]. Second, under a reasonable bound on agent heterogeneity, we found that for sufficiently small step-sizes, the performance guarantees of the decentralized strategy essentially match those for the centralized framework, implying that even in the absence of central aggregation of data or parameter estimates, decentralized strategies can yield competitive performance in terms of their second-order guarantees, a fact that is well established for strongly-convex costs, but only recently has received attention in the nonconvex setting.

References

[1] K. G. Murty and Santosh N. Kabadi, "Some NP-complete problems in quadratic and nonlinear programming," *Mathematical Programming*, vol. 39, no. 2, pp. 117–129, Jun 1987.

[2] S. Vlaski and A. H. Sayed, "Second-order guarantees of stochastic gradient descent in non-convex optimization," *available as arXiv:1908.07023*, August 2019.

[3] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments – Part I: Agreement at a linear rate," *submitted for publication, available as arXiv:1907.01848*, July 2019.

[4] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments – Part II: Polynomial escape from saddle-points," *submitted for publication, available as arXiv:1907.01849*, July 2019.

[5] 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. MR4165946

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436 EP –, 05 2015. MR1110084

[7] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug 2009.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[9] Y. Nesterov, *Introductory Lectures on Convex Programming Volume I: Basic Course*, Springer, 1998.

[10] D. Bertsekas and J. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM Journal on Optimization*, vol. 10, no. 3, pp. 627–642, 2000.

[11] S. J. Reddi, A. Hefny, S. Sra, B. Póczós, and A. Smola, "Stochastic variance reduction for nonconvex optimization," in *Proc. of ICML*, New York, NY, USA, 2016, pp. 314–323.

[12] 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.

[13] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, Aug. 2017. MR2579916

[14] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case

study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems 30*, pp. 5330–5340. 2017.

[15] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "d^2: Decentralized training over decentralized data," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, vol. 80, pp. 4848–4856. MR2478070

[16] 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. MR2654432

[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] Y. Nesterov and B.T. Polyak, "Cubic regularization of newton method and its global performance," *Mathematical Programming*, vol. 108, no. 1, pp. 177–205, Aug 2006.

[19] 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. MR3352511

[20] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *29th Annual Conference on Learning Theory*, New York, 2016, pp. 1246–1257. MR3708874

[21] R. Ge, J. D. Lee, and T. Ma, "Matrix completion has no spurious local minimum," in *Advances in Neural Information Processing Systems 29*, pp. 2973–2981. Curran Associates, Inc., 2016. MR3912283

[22] R. Ge, C. Jin, and Y. Zheng, "No spurious local minima in nonconvex low rank problems: A unified geometric analysis," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1233–1242. MR2932818

[23] K. Kawaguchi, "Deep learning without poor local minima," in *Advances in Neural Information Processing Systems 29*, pp. 586–594. 2016.

[24] F. E. Curtis, D. P. Robinson, and M. Samadi, "A trust region algorithm with a worst-case iteration complexity of $o(\epsilon^{-3/2})$ for nonconvex optimization," *Mathematical Programming*, vol. 162, pp. 1–32, 2017. MR3189404

[25] N. Tripuraneni, M. Stern, C. Jin, J. Regier, and M. I. Jordan, "Stochastic cubic regularization for fast nonconvex optimization," in *Proceedings International Conference on Neural Information Processing Systems*, USA, 2018, pp. 2904–2913.

[26] C. Fang, C. J. Li, Z. Lin, and T. Zhang, "SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator," in *Proc. of NIPS*, pp. 689–699. Montreal, Canada, 2018. MR3359883

[27] Z. Allen-Zhu and Y. Li, "NEON2: Finding local minima via first-order oracles," in *Proc. of NIPS*, pp. 3716–3726. Montreal, Canada, Dec. 2018. MR3340785

[28] Z. Allen-Zhu, "Natasha 2: Faster non-convex optimization than SGD," in *Proc. of NIPS*, pp. 2675–2686. Montreal, Canada, Dec. 2018. MR3916294

[29] S. S. Du, C. Jin, J. D. Lee, M. I. Jordan, B. Póczos, and A. Singh, "Gradient descent can take exponential time to escape saddle points," in *Proceedings International Conference on Neural Information Processing Systems*, 2017, pp. 1067–1077. MR2978290

[30] A. Daneshmand, G. Scutari and V. Kungurtsev, "Second-order guarantees of distributed gradient algorithms," *available as arXiv:1809.08694*, Sep. 2018.

[31] S. Gelfand and S. Mitter, "Recursive stochastic algorithms for global optimization in $\mathbb{R}^d$," *SIAM Journal on Control and Optimization*, vol. 29, no. 5, pp. 999–1018, 1991.

[32] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in *Proc. of ICML*, Sydney, Australia, Aug. 2017, pp. 1724–1732. MR3352512

[33] C. Fang, Z. Lin and T. Zhang, "Sharp analysis for nonconvex SGD escaping from saddle points," *available as arXiv:1902.00247*, Feb. 2019.

[34] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade and M. I. Jordan, "Stochastic gradient descent escapes saddle points efficiently," *available as arXiv:1902.04811*, Feb. 2019.

[35] H. Daneshmand, J. Kohler, A. Lucchi and T. Hofmann, "Escaping saddles with stochastic gradients," *available as arXiv:1803.05999*, March 2018.

[36] B. Swenson, S. Kar, H. V. Poor and J. M. F. Moura, "Annealing for distributed global optimization," *available as arXiv:1903.07258*, March 2019.

[37] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Advances in Neural Information Processing Systems*, vol. 23, pp. 2595–2603, 2010.

[38] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *Journal of Machine Learning Research*, vol. 10, pp. 2899–2934, 2009.

[39] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al., "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 20–22 April 2017.

[40] 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.

[41] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, April 2010.

[42] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.

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

[44] 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.

[45] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, July 2018.

[46] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning—Part I: Algorithm development," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 708–723, Feb 2019.

[47] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, March 2012.

[48] K. I. Tsianos and M. G. Rabbat, "Distributed dual averaging for convex optimization under communication delays," in *Proc. American Control Conference (ACC)*, Montreal, Canada, June 2012, pp. 1067–1072.

[49] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, April 2014.

[50] M. Jaggi, V. Smith, M. Takáč, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *Proc. International Conference on Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 3068–3076.

[51] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, Aug 2015.

[52] D. Jakovetić, J. M. F. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented Lagrangian algorithms," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 922–936, April 2015.

[53] D. Jakovetić, "A unification and generalization of exact distributed first-order methods," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 31–46, March 2019.

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

[55] 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.

[56] S. Vlaski and A. H. Sayed, "Linear speedup in saddle-point escape for decentralized non-convex optimization," *submitted for publication, available as arXiv:1910.13852*, July 2019.

[57] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks – Part II: Performance analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3518–3548, June 2015.

Stefan Vlaski
Institute of Electrical Engineering
École Polytechnique Fédérale de Lausanne
Lausanne
Switzerland
E-mail address: stefan.vlaski@epfl.ch

Ali H. Sayed
Institute of Electrical Engineering
École Polytechnique Fédérale de Lausanne
Lausanne
Switzerland
E-mail address: ali.sayed@epfl.ch

Received March 15, 2020