

EXACT SUBSPACE DIFFUSION FOR DECENTRALIZED MULTITASK LEARNING

Shreya Wadehra*, Roula Nassif† and Stefan Vlaski*

* Department of Electrical and Electronic Engineering, Imperial College London, UK

† I3S Laboratory, Université Côte d’Azur, France

ABSTRACT

Classical paradigms for distributed learning, such as federated or decentralized gradient descent, employ consensus mechanisms to enforce homogeneity among agents. While these strategies have proven effective in i.i.d. scenarios, they can result in significant performance degradation when agents follow heterogeneous objectives or data. Distributed strategies for multitask learning, on the other hand, induce relationships between agents in a more nuanced manner, and encourage collaboration without enforcing consensus. We develop a generalization of the exact diffusion algorithm for subspace constrained multitask learning over networks, and derive an accurate expression for its mean-squared deviation when utilizing noisy gradient approximations. We verify numerically the accuracy of the predicted performance expressions, as well as the improved performance of the proposed approach over alternatives based on approximate projections.

Index Terms— Decentralized learning, federated learning, multitask learning, bias-correction, subspace constraints.

1. INTRODUCTION

We consider a collection of K agents, indexed by k , where each agent is equipped with a local risk $J_k(w_k)$ of the form:

$$J_k(w_k) = \mathbb{E}Q_k(w_k; \mathbf{x}_k) \quad (1)$$

Here, $w_k \in \mathbb{R}^M$ denotes some parameterization chosen by agent k and \mathbf{x}_k describes locally available data, modeled as a random variable. The loss $Q_k(w_k; \mathbf{x}_k)$ then measures how well w_k fits to the data \mathbf{x}_k , and the risk $J_k(w_k)$ measures the expected loss.

In a networked setting, where agents have the ability to communicate and exchange information, we may then consider various learning paradigms. As a baseline, in a non-cooperative mode of operation, agents pursue locally optimal models independently by minimizing their local risks:

$$w_k^o \triangleq \arg \min_{w_k \in \mathbb{R}^M} J_k(w_k) \quad (2)$$

An alternative to non-cooperative approaches is the consensus optimization problem [1–6]:

$$w^o \triangleq \arg \min_{w \in \mathbb{R}^M} \frac{1}{K} \sum_{k=1}^K J_k(w) \quad (3)$$

In contrast to the independent learning objectives in (2), local objectives are coupled in (3) into a single task of coming to consensus on an optimal average model w^o . For this reason, the problem of consensus optimization is also referred to as single-task learning. Depending on restrictions on the flow of information, solutions to (3) may be pursued using a number of different learning strategies, ranging from centralized or parallel [2] to federated [5] or fully-decentralized structures [1, 3, 4, 7–12].

When agents are homogeneous, meaning that their local data \mathbf{x}_k are identically and independently distributed, and they employ the same loss $Q_k(\cdot; \cdot)$, it follows that the models w_k^o and w^o coincide. In such settings, K -fold improvement in performance has been established for a number of different algorithms, and a number of different performance metrics, including the mean-squared deviation for convex risks [13] or first-order stationarity [14] and second-order stationarity [15] for non-convex risks.

In heterogeneous settings, performance trade-offs become more nuanced, since solving the consensus optimization problem (3) induces a bias relative to locally optimal models (2). Depending on the level of heterogeneity in the network, this bias can outweigh any benefit gained from cooperation, and result in a consensus model w^o , which is optimal on average, but performs poorly on any local objective. This observation has motivated the inclusion of tools from multitask learning [16] in both federated [17] and decentralized settings [18–24]. Strategies for distributed multitask learning couple local objectives without enforcing strict consensus. While various models have been considered in the literature, we will focus in this work on subspace constrained multitask learning problems of the form [18, 22, 23]:

$$w^* \triangleq \arg \min_{w \in \mathcal{R}(\mathcal{U})} \sum_{k=1}^K J_k(w_k) \quad (4)$$

Here, we denote by $\mathcal{R}(\mathcal{U})$ the range space of some matrix $\mathcal{U} \in \mathbb{R}^{KM \times P}$ with full column-rank and introduce the net-

Emails: shreya.wadehra22@imperial.ac.uk, roula.nassif@unice.fr, s.vlaski@imperial.ac.uk

work quantity \mathcal{W} , which is obtained by stacking the individual models w_k as in:

$$\mathcal{W} \triangleq \text{col} \{w_k\} \quad (5)$$

Note that choosing $\mathcal{U} = I_{KM}$ results in $\mathcal{R}(\mathcal{U}) = \mathbb{R}^{KM}$, and hence

$$\mathcal{W}^* \triangleq \arg \min_{\mathcal{W} \in \mathbb{R}^{KM}} \sum_{k=1}^K J_k(w_k) \quad (6)$$

decouples into the independent local optimization problems of (2). In this case, \mathcal{W}^* coincides with $\mathcal{W}^o \triangleq \text{col} \{w_k^o\}$, which is obtained by stacking the solutions w_k^o of (2). On the other hand, if we let $\mathcal{U} = \mathbf{1}_K \otimes I_M$, it can be verified [22] that:

$$\mathcal{W} \in \mathcal{R}(\mathbf{1}_K \otimes I_M) \iff w_k = w_\ell \text{ for all } k, \ell. \quad (7)$$

Hence, in this case (4) reduces to the consensus optimization problem (3). Other choices of \mathcal{U} result in alternative task-relationship priors by restricting models w_k contained in \mathcal{W} to lie in a lower-dimensional linear subspace spanned by \mathcal{U} . For example, setting \mathcal{U} to the leading eigenvectors of the graph Fourier transform of an underlying graph [25] results in band-limited recovery, while other choices of \mathcal{U} can be used to encode overlap or pairwise linear constraints between agents. We refer the reader to [22,23] for further examples and details on how to match the choice of subspace constraint to the underlying signal processing, learning or optimization problem.

2. PROBLEM FORMULATION AND RELATED WORKS

2.1. Network Model

Each agent in the network is represented by a node in a graph, where edges between agents represent communication links, meaning that information may be exchanged between this pair of agents. We denote by \mathcal{N} the set of all agents, and by \mathcal{N}_k the neighborhood of agent k . In other words, $\ell \in \mathcal{N}_k$ implies that there is an edge from agent ℓ to agent k . We will assume the graph to be undirected, meaning that $\ell \in \mathcal{N}_k \iff k \in \mathcal{N}_\ell$.

2.2. Approximate Projection-Based Algorithms

We begin by briefly reviewing the derivations and algorithms of [22, 23], which are closely related to our proposed algorithm, before pointing out their limitations. Since $\mathcal{R}(\mathcal{U})$ denotes a linear subspace, its projection can be determined in closed form, and is expressed as:

$$\mathcal{P}_\mathcal{U} = \mathcal{U}(\mathcal{U}^\top \mathcal{U})^{-1} \mathcal{U}^\top \quad (8)$$

It is then, at least in principle, possible to pursue \mathcal{W}^* in (4) via projected gradient descent, which takes the form:

$$w_i = \mathcal{P}_\mathcal{U} (w_{i-1} - \mu \nabla \mathcal{J}(w_{i-1})) \quad (9)$$

where we defined:

$$\nabla \mathcal{J}(w_{i-1}) \triangleq \text{col} \{ \nabla J_k(w_{k,i-1}) \} \quad (10)$$

Two factors limit the applicability of (9) in networked learning environments. First, the projection matrix $\mathcal{P}_\mathcal{U}$ is in general dense, meaning that an iteration of (9) would require central aggregation of the local models $w_{k,i-1}$ contained in w_{i-1} . Second, a step of projected gradient descent requires local access to the exact gradients $\nabla J_k(w_{k,i-1}) = \nabla \mathbb{E}Q(w_{k,i-1}; \mathbf{x}_k)$, which in turn requires knowledge of the distribution of \mathbf{x}_k . This is unavailable when learning from finite or streaming realizations of data. The first limitation is addressed in both [22] and [23] by replacing the dense projection matrix by a block-sparse approximation \mathcal{A} , which satisfies:

$$\lim_{i \rightarrow \infty} \mathcal{A}^i = \mathcal{P}_\mathcal{U} \quad (11)$$

$$A_{\ell k} \triangleq [\mathcal{A}]_{\ell k} = 0, \text{ if } \ell \notin \mathcal{N}_k \quad (12)$$

Here, $[\mathcal{A}]_{\ell k}$ denotes ℓk -th block of \mathcal{A} of dimension $M \times M$. It has been shown that equivalent conditions on \mathcal{A} are given by [18, 22, 23]:

$$\mathcal{A} \mathcal{P}_\mathcal{U} = \mathcal{P}_\mathcal{U} \quad (13)$$

$$\mathcal{P}_\mathcal{U} \mathcal{A} = \mathcal{P}_\mathcal{U} \quad (14)$$

$$\lambda_\mathcal{A} \triangleq \rho(\mathcal{P}_\mathcal{U} - \mathcal{A}) < 1 \quad (15)$$

Valid choices of \mathcal{A} can be constructed for a given \mathcal{U} and sufficiently connected network topology by solving a convex optimization problem [22, 23]. For simplicity, we will assume throughout that \mathcal{A} is constructed to be symmetric. If not, we could simply make the replacement $\mathcal{A} \leftarrow \frac{1}{2}(\mathcal{A} + \mathcal{A}^\top)$. The algorithm of [22] is obtained by directly replacing $\mathcal{P}_\mathcal{U}$ by \mathcal{A} in (9) to obtain:

$$w_i = \mathcal{A} (w_{i-1} - \mu \nabla \mathcal{J}(w_{i-1})) \quad (16)$$

We may return to node-level quantities by exploiting the block-structures of $\{\mathcal{A}, w_{i-1}, \mathcal{J}(w_{i-1})\}$ to obtain:

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} A_{\ell k} (w_{\ell,i-1} - \mu \nabla J_\ell(w_{\ell,i-1})) \quad (17)$$

The limiting condition (11) ensures that for small step-sizes μ , recursion (16) approximates (9), while the sparsity condition (12) on the other hand ensures that (16) or (17) can be implemented by relying only on communication exchanges between neighboring agents. The DiSPO algorithm of [23], on the other hand, applies \mathcal{A} only to w_{i-1} , resulting in:

$$w_i = \mathcal{A} w_{i-1} - \mu \nabla \mathcal{J}(w_{i-1}) \quad (18)$$

or in local quantities:

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} A_{\ell k} w_{\ell,i-1} - \mu \nabla J_k(w_{k,i-1}) \quad (19)$$

We remark that (17) and (19) can be viewed as generalizations of the Adapt-then-Combine (ATC) diffusion algorithm [4, 7] and distributed gradient descent [1] respectively, where the typical scalar combination weights are replaced by linear transformations $A_{\ell k}$. It is precisely these linear transformations that allow the decentralized algorithms (17) and (19) to solve generic subspace constrained problems of the form (4), rather than consensus problems of the form (3).

The second limitation is addressed in [22] by replacing true gradients in (16) by stochastic approximations $\widehat{\nabla} J_k(\mathbf{w}_{k,i-1})$, based on data available at time i . This results in:

$$\mathbf{w}_i = \mathcal{A} \left(\mathbf{w}_{i-1} - \mu \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) \right) \quad (20)$$

or

$$\mathbf{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} A_{\ell k} \left(\mathbf{w}_{\ell,i-1} - \mu \widehat{\nabla} J_{\ell}(\mathbf{w}_{\ell,i-1}) \right) \quad (21)$$

where we now employ bold font for \mathbf{w}_i or $\mathbf{w}_{k,i}$ to emphasize that iterates will be random.

2.3. Primal-Dual Algorithms

Both ATC-diffusion and the decentralized gradient descent algorithm for consensus optimization are known to exhibit a fixed-point bias [26, 27]. This has motivated a number of approaches for bias-correction in the context of consensus optimization using arguments based on Lagrangian duality and gradient tracking [8–10]. Although bias-correction was originally motivated in deterministic settings, its potential benefit has also been established in stochastic settings [28].

A similar bias has been observed in the context of subspace constrained optimization for (18) in [23], and can be verified for (16) using similar arguments. These considerations motivate the development of bias-corrected strategies for decentralized subspace constrained optimization. A bias-corrected version of DiSPO for deterministic optimization, termed EDiSPO, is provided in [23] by adjusting the arguments that lead to EXTRA [8]. The recent work [29] provides generalizations of a large number of bias-corrected algorithms including exact diffusion for consensus optimization to the subspace constrained setting and provides convergence and sensitivity analysis in the presence of i.i.d. perturbations using the integral quadratic constraint framework. In relation to these related works, we make the following contributions:

- We derive an exact subspace diffusion algorithm by extending the incremental arguments of [9] to the subspace constrained setting. Incremental arguments of this type have been shown to yield wider stability ranges [7, 9].
- We allow for stochastic gradient approximations in lieu of true gradients, which induces gradient noise. In contrast to the perturbations in [29], this type of gradient noise is Markovian, rather than independent and identically distributed.

- We allow for multiple local updates to take place in between every communication exchange. This flexibility can improve communication efficiency in federated and decentralized settings [5, 12].
- When agents perform a single update per exchange, we derive an expression for the limiting mean-squared deviation of the proposed algorithm, which matches the centralized benchmark, and show numerically that it approximates the true performance over a wide range of conditions.

3. ALGORITHM DEVELOPMENT

The derivation essentially mirrors that of [9], after accounting for the more general subspace constraints, and allowing for multiple primal updates along the aggregate objective $\sum_{k=1}^K J_k(w_k)$. To this end, note that (4) is equivalent to:

$$\mathbf{w}^* \triangleq \arg \min_{\mathbf{w} \in \mathbb{R}^{KM}} \sum_{k=1}^K J_k(w_k) \text{ s.t. } (I_{KM} - \mathcal{P}_U) \mathbf{w} = 0 \quad (22)$$

As long as \mathcal{A} satisfies (11)–(12), this is further equivalent to [23]:

$$\mathbf{w}^* \triangleq \arg \min_{\mathbf{w} \in \mathbb{R}^{KM}} \sum_{k=1}^K J_k(w_k) \text{ s.t. } (I_{KM} - \mathcal{A}) \mathbf{w} = 0 \quad (23)$$

Analogously to [9], we introduce the augmented Lagrangian:

$$\mathcal{L}(\mathbf{w}, \lambda) \triangleq \sum_{k=1}^K J_k(w_k) + \frac{1}{\mu} \lambda^\top \mathcal{B} \mathbf{w} + \frac{1}{4\mu} \mathbf{w}^\top (I_{KM} - \mathcal{A}) \mathbf{w} \quad (24)$$

Here, \mathcal{B} denotes the square root of the matrix $\frac{1}{2}(I_{KM} - \mathcal{A}) = \mathcal{B} \cdot \mathcal{B}$, which exists as long as \mathcal{A} is symmetric with spectral radius bounded by one. We can pursue a saddle-point to the augmented Lagrangian via incremental stochastic gradient descent-ascent, initializing $\psi_{i,0} = \mathbf{w}_{i-1}$ and letting:

$$\psi_{i,e} = \psi_{i,e-1} - \frac{\mu}{E} \widehat{\nabla} \mathcal{J}(\psi_{i,e-1}) \text{ for } e = 1, \dots, E \quad (25)$$

$$\mathbf{w}_i = \bar{\mathcal{A}} \psi_{i,E} - \mathcal{B} \lambda_{i-1} \quad (26)$$

$$\lambda_i = \lambda_{i-1} + \mathcal{B} \mathbf{w}_i \quad (27)$$

where we defined $\bar{\mathcal{A}} = \frac{1}{2}(I_{KM} + \mathcal{A})$ and E denotes the number of local updates. Evaluating (26) at time $i-1$ yields:

$$\mathbf{w}_{i-1} = \bar{\mathcal{A}} \psi_{i-1,E} - \mathcal{B} \lambda_{i-2} \quad (28)$$

Subtracting (28) from (26), and using (27):

$$\begin{aligned} & \mathbf{w}_i - \mathbf{w}_{i-1} \\ &= \bar{\mathcal{A}} \psi_{i,E} - \bar{\mathcal{A}} \psi_{i-1,E} - \mathcal{B} (\lambda_{i-1} - \lambda_{i-2}) \\ &= \bar{\mathcal{A}} \psi_{i,E} - \bar{\mathcal{A}} \psi_{i-1,E} - \mathcal{B}^2 \mathbf{w}_{i-1} \\ &= \bar{\mathcal{A}} \psi_{i,E} - \bar{\mathcal{A}} \psi_{i-1,E} - \frac{1}{2} \mathbf{w}_{i-1} + \frac{1}{2} \mathcal{A} \mathbf{w}_{i-1} \end{aligned} \quad (29)$$

After rearranging, we have:

$$\mathbf{w}_i = \bar{\mathcal{A}} (\mathbf{w}_{i-1} + \boldsymbol{\psi}_{i,E} - \boldsymbol{\psi}_{i-1,E}) \quad (30)$$

Upon returning to local quantities, we obtain the proposed exact subspace diffusion algorithm with local updates in Algorithm 1. We remark that setting $E = 1$ and $A_{\ell k} = a_{\ell k} I_M$, we recover the exact diffusion algorithm of [9], which justifies the name.

Algorithm 1: Exact Subspace Diffusion

Initialize $\mathbf{w}_{k,0}$ arbitrary and $\boldsymbol{\psi}_{k,0,E} = \mathbf{w}_{k,0}$. Set:

$$\bar{\mathcal{A}} = \frac{1}{2} (\mathcal{A} + I_{KM}) \quad (31)$$

For $i \geq 1$, set $\boldsymbol{\psi}_{k,i,0} = \mathbf{w}_{k,i-1}$ and perform E local updates for $e = 1, \dots, E$:

$$\boldsymbol{\psi}_{k,i,e} = \boldsymbol{\psi}_{k,i,e-1} - \frac{\mu}{E} \widehat{\nabla} J_k(\boldsymbol{\psi}_{k,i,e-1}) \quad (32)$$

Correct:

$$\boldsymbol{\phi}_{k,i} = \mathbf{w}_{k,i-1} + \boldsymbol{\psi}_{k,i,E} - \boldsymbol{\psi}_{k,i-1,E} \quad (33)$$

Exchange and transform:

$$\mathbf{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} \bar{A}_{\ell k} \boldsymbol{\phi}_{\ell,i} \quad (34)$$

4. CONVERGENCE ANALYSIS

For simplicity, in this section, we will restrict ourselves to single local updates $E = 1$. We can then write (25)–(27) compactly as:

$$\mathbf{w}_i = \bar{\mathcal{A}} \mathbf{w}_{i-1} - \mu \bar{\mathcal{A}} \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) - \mu \mathcal{B} \boldsymbol{\lambda}_{i-1} \quad (35)$$

$$\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} + \mathcal{B} \mathbf{w}_i \quad (36)$$

Throughout this section, we will be interested in quantifying the mean-squared deviation (MSD) of the network iterates \mathbf{w}_i around \mathbf{w}^* , defined by (4), namely $\mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i\|^2$. The network iterates \mathbf{w}_i satisfy the decomposition:

$$\mathbf{w}_i = \mathcal{P}_U \mathbf{w}_i + (I_{KM} - \mathcal{P}_U) \mathbf{w}_i = \mathbf{w}_i^U + \mathbf{w}_i^{\perp U} \quad (37)$$

Here, we defined:

$$\mathbf{w}_i^U \triangleq \mathcal{P}_U \mathbf{w}_i \quad (38)$$

$$\mathbf{w}_i^{\perp U} \triangleq (I_{KM} - \mathcal{P}_U) \mathbf{w}_i \quad (39)$$

Since for any projection $\mathcal{P}_U = \mathcal{P}_U^2$, it can be readily verified that \mathbf{w}_i^U and $\mathbf{w}_i^{\perp U}$ are orthogonal. Using this, and the fact

that $\mathbf{w}^* \in \mathcal{R}(U)$, it follows that:

$$\begin{aligned} \mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i\|^2 &= \mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i^U - \mathbf{w}_i^{\perp U}\|^2 \\ &= \mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i^U\|^2 + \mathbb{E} \|\mathbf{w}_i^{\perp U}\|^2 \end{aligned} \quad (40)$$

We may then equivalently express the mean-squared deviation $\mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i\|^2$ by instead quantifying the orthogonal contributions $\mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i^U\|^2$ and $\mathbb{E} \|\mathbf{w}_i^{\perp U}\|^2$, which is a common theme in the study of decentralized learning algorithms [7, 22, 23, 26]. Applying the projector \mathcal{P}_U to (35) yields:

$$\begin{aligned} \mathbf{w}_i^U &= \mathcal{P}_U \bar{\mathcal{A}} \mathbf{w}_{i-1} - \mu \mathcal{P}_U \bar{\mathcal{A}} \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) - \mu \mathcal{P}_U \mathcal{B} \boldsymbol{\lambda}_{i-1} \\ &\stackrel{(a)}{=} \mathcal{P}_U \left(\mathbf{w}_{i-1} - \mu \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) \right) \\ &= \mathbf{w}_{i-1}^U - \mu \mathcal{P}_U \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) \\ &= \mathcal{P}_U \left(\mathbf{w}_{i-1}^U - \mu \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) \right) \end{aligned} \quad (41)$$

where (a) follows from the spectral structure of $\bar{\mathcal{A}}$, induced by the conditions (11)–(12) [18, 22, 23], ensuring that:

$$\mathcal{P}_U \bar{\mathcal{A}} = \frac{1}{2} \mathcal{P}_U (I_{KM} + \mathcal{A}) = \mathcal{P}_U \quad (42)$$

$$\mathcal{P}_U \mathcal{B} = \mathcal{P}_U \left(\frac{1}{2} (I_{KM} - \mathcal{A}) \right)^{\frac{1}{2}} = 0 \quad (43)$$

In interpreting recursion (41), it is useful to consider a centralized benchmark. In the absence of communication constraints, one could pursue a solution to (4) via projected stochastic gradient descent, yielding the recursion:

$$\mathbf{w}_i^{\text{cent}} = \mathcal{P}_U \left(\mathbf{w}_{i-1}^{\text{cent}} - \mu \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}^{\text{cent}}) \right) \quad (44)$$

Comparing (44) to (41), we observe that the recursions are almost identical, except that the stochastic gradients in (41) are evaluated at \mathbf{w}_{i-1} instead of \mathbf{w}_{i-1}^U . If $\mathbf{w}_{i-1} \approx \mathbf{w}_{i-1}^U$, and under suitable smoothness conditions on $\widehat{\nabla} \mathcal{J}(\cdot)$, it is then reasonable to expect recursion (41) to track the centralized benchmark (44). In light of (37), the deviation $\mathbf{w}_i - \mathbf{w}_i^U$ is given by $\mathbf{w}_i^{\perp U} \triangleq (I_{KM} - \mathcal{P}_U) \mathbf{w}_i$. Applying $(I_{KM} - \mathcal{P}_U)$ to (35), we find:

$$\begin{aligned} \mathbf{w}_i^{\perp U} &= (I_{KM} - \mathcal{P}_U) \bar{\mathcal{A}} \mathbf{w}_{i-1} - \mu (I_{KM} - \mathcal{P}_U) \bar{\mathcal{A}} \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) \\ &\quad - \mu (I_{KM} - \mathcal{P}_U) \mathcal{B} \boldsymbol{\lambda}_{i-1} \\ &\stackrel{(14)}{=} (\bar{\mathcal{A}} - \mathcal{P}_U) \mathbf{w}_{i-1} - \mu (\bar{\mathcal{A}} - \mathcal{P}_U) \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) - \mu \mathcal{B} \boldsymbol{\lambda}_{i-1} \\ &\stackrel{(13)}{=} (\bar{\mathcal{A}} - \mathcal{P}_U) (I_{KM} - \mathcal{P}_U) \mathbf{w}_{i-1} - \mu (\bar{\mathcal{A}} - \mathcal{P}_U) \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) \\ &\quad - \mu \mathcal{B} \boldsymbol{\lambda}_{i-1} \\ &= (\bar{\mathcal{A}} - \mathcal{P}_U) \mathbf{w}_{i-1}^{\perp U} - \mu (\bar{\mathcal{A}} - \mathcal{P}_U) \widehat{\nabla} \mathcal{J}(\mathbf{w}_{i-1}) - \mu \mathcal{B} \boldsymbol{\lambda}_{i-1} \end{aligned} \quad (45)$$

In light of (15), we have:

$$\begin{aligned}\lambda_{\bar{\mathcal{A}}} &\triangleq \rho (\bar{\mathcal{A}} - \mathcal{P}_U) = \rho \left(\frac{1}{2} (I_{KM} + \mathcal{A}) - \mathcal{P}_U \right) \\ &\leq \frac{1}{2} \rho (I_{KM} - \mathcal{P}_U) + \frac{1}{2} \rho (\mathcal{A} - \mathcal{P}_U) \\ &\leq \frac{1}{2} + \frac{1}{2} \lambda_{\mathcal{A}} < 1\end{aligned}\quad (46)$$

It follows that the recursion (45) in the deviation $\mathbf{w}_i^{\perp U}$ is driven by a stable matrix. Some care needs to be taken, since the recursions (41) and (45) are coupled through the driving term $\widehat{\nabla \mathcal{J}}(\mathbf{w}_{i-1})$, and (45) is additionally driven by the dual variable λ_{i-1} .

4.1. Stability

We introduce the following common modeling conditions on the objectives $J_k(\cdot)$ and gradient approximations $\widehat{\nabla J}_k(\cdot)$ [7, 22, 28].

Assumption 1 (Conditions on $J_k(\cdot)$). *Each local objective $J_k(\cdot)$ is ν_k -strongly convex with δ_k -Lipschitz gradients, meaning that for every $x, y \in \mathbb{R}^M$, it holds that:*

$$\begin{aligned}(\nabla J_k(x) - \nabla J_k(y))^\top (x - y) &\geq \nu_k \|x - y\|^2 \quad (47) \\ \|\nabla J_k(x) - \nabla J_k(y)\| &\leq \delta_k \|x - y\| \quad (48)\end{aligned}$$

for some $0 < \nu_k < \delta_k$. \square

Assumption 2 (Conditions on $\widehat{\nabla J}_k(\cdot)$). *Define the local gradient noise process:*

$$\mathbf{s}_{k,i}(\mathbf{w}_{k,i-1}) \triangleq \widehat{\nabla J}_k(\mathbf{w}_{k,i-1}) - \nabla J_k(\mathbf{w}_{k,i-1}) \quad (49)$$

The gradient noise is zero-mean after conditioning on the current iterate:

$$\mathbb{E} \{ \mathbf{s}_{k,i}(\mathbf{w}_{k,i-1}) \mid \mathbf{w}_{k,i-1} \} = 0 \quad (50)$$

Furthermore, its variance is bounded according to:

$$\mathbb{E} \{ \|\mathbf{s}_{k,i}(\mathbf{w}_{k,i-1})\|^2 \mid \mathbf{w}_{k,i-1} \} \leq \beta_k^2 \|\mathbf{w}_k^* - \mathbf{w}_{k,i-1}\|^2 + \sigma_k^2 \quad (51)$$

Finally, the gradient noise processes between agents $k \neq \ell$ are uncorrelated after conditioning on current iterates:

$$\mathbb{E} \{ \mathbf{s}_{k,i}(\mathbf{w}_{k,i-1}) \mathbf{s}_{\ell,i}(\mathbf{w}_{\ell,i-1})^\top \mid \mathbf{w}_{k,i-1}, \mathbf{w}_{\ell,i-1} \} = 0 \quad (52)$$

\square

A common gradient approximation is given by $\widehat{\nabla J}_k(\mathbf{w}_{k,i-1}) = \nabla Q(\mathbf{w}_{k,i-1}; \mathbf{x}_{k,i})$, where $\mathbf{x}_{k,i}$ denotes the sample available to agent k at time i . Alternative constructions, such as mini-batch or asynchronous updates are possible as well. In those cases, and for many loss functions $Q(\cdot; \cdot)$ arising in learning problems, the conditions in Assumption 2 can be verified to hold — we refer the reader to [6, 30] for details. We can then establish the following stability result.

Lemma 1 (Mean-square stability). *Under Assumptions 1–2, and for symmetric \mathcal{A} satisfying (13)–(15), there exists a step-size μ small enough, so that the exact subspace diffusion recursions (32)–(34) are stable in the mean-square sense, and:*

$$\limsup_{i \rightarrow \infty} \mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i\|^2 = O(\mu) \quad (53)$$

Proof. The proof continues from recursions (41) and (45) using arguments similar to [9, 28] and is omitted due to space limitations. We verify the claim numerically in Section 5. \square

The $O(\cdot)$ notations in (53) denotes that the dependence of $\limsup_{i \rightarrow \infty} \mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i\|^2$ is approximately linear in μ for small step-sizes, or more precisely that

$$\lim_{\mu \rightarrow 0} \frac{1}{\mu} \limsup_{i \rightarrow \infty} \mathbb{E} \|\mathbf{w}^* - \mathbf{w}_i\|^2 \quad (54)$$

tends towards a finite constant as μ tends to zero. To develop a more clear understanding of the performance of the proposed algorithm, we derive an expression for this constant in the next section.

4.2. Performance

To provide a more granular understanding of the steady-state performance of the proposed algorithm, we will need to introduce additional regularity conditions on the objectives and gradient approximations. Analogous conditions have been used before when deriving performance expressions of single-task learning algorithms [7], multitask learning algorithms [21, 31], or when establishing the ability of stochastic gradient algorithms to escape from saddle-points [32].

Assumption 3 (Higher-order smoothness). *The local objectives $J_k(\cdot)$ are twice-differentiable, with Lipschitz continuous Hessian matrix $\nabla^2 J_k(\cdot)$ around \mathbf{w}_k^* satisfying for all x :*

$$\|\nabla^2 J_k(x) - \nabla^2 J_k(\mathbf{w}_k^*)\| \leq \kappa_k \|x - \mathbf{w}_k^*\| \quad (55)$$

We define as H_k^* the Hessian at the optimal solution of (4), namely $H_k^* \triangleq \nabla^2 J_k(\mathbf{w}_k^*)$, and $\mathcal{H}^* \triangleq \text{diag} \{ H_k^* \}$. As we will see, \mathcal{H}^* will play a role in the final performance expression. \square

Assumption 4 (Higher-order gradient noise conditions). *The fourth moment of the gradient noise process (49) also satisfies a relative bound of the form:*

$$\mathbb{E} \{ \|\mathbf{s}_{k,i}(\mathbf{w}_{k,i-1})\|^4 \mid \mathbf{w}_{k,i-1} \} \leq \beta_{4,k}^4 \|\mathbf{w}_k^* - \mathbf{w}_{k,i-1}\|^4 + \sigma_{4,k}^4 \quad (56)$$

We introduce the gradient noise covariance:

$$R_{s,k}(\mathbf{w}_{k,i-1}) \triangleq \mathbb{E} \{ \mathbf{s}_{k,i}(\mathbf{w}_{k,i-1}) \mathbf{s}_{k,i}(\mathbf{w}_{k,i-1})^\top \mid \mathbf{w}_{k,i-1} \} \quad (57)$$

The gradient noise covariance is also assumed to satisfy a smoothness condition around w_k^* :

$$\|R_{s,k}(x) - R_{s,k}(w_k^*)\| \leq \kappa \|x - w_k^*\|^\gamma \quad (58)$$

where $\kappa_k \geq 0$ and $0 < \gamma \leq 4$. We again define $R_{s,k}^* \triangleq R_{s,k}(w_k^*)$ and $\mathcal{R}_s^* \triangleq \text{diag}\{R_{s,k}^*\}$. \square

Given these assumptions, we can then provide a precise characterization of the limiting behavior of the algorithm for small step-sizes.

Theorem 1 (Mean-squared deviation of exact subspace diffusion). *Under Assumptions 1–4, the mean-squared deviation satisfies:*

$$\limsup_{i \rightarrow \infty} \mathbb{E} \|\mathcal{W}^* - \mathcal{W}_i\|^2 = \frac{\mu^2}{K} \text{Tr} \left(\sum_{n=0}^{\infty} \mathcal{C}^n \mathcal{Y} \mathcal{C}^n \right) + o(\mu) \quad (59)$$

where $o(\mu)$ denotes a higher-order term in μ and:

$$\mathcal{C} = \mathcal{A} (I_{KM} - \mu \mathcal{H}^*) \quad (60)$$

$$\mathcal{Y} = \mathcal{A} \mathcal{R}_s^* \mathcal{A} \quad (61)$$

Asymptotically, for small μ , this may be approximated by:

$$\begin{aligned} & \mu \lim_{\mu \rightarrow 0} \frac{1}{\mu} \limsup_{i \rightarrow \infty} \mathbb{E} \|\mathcal{W}^* - \mathcal{W}_i\|^2 \\ &= \frac{\mu}{2K} \text{Tr} \left((U^\top \mathcal{H}^* U)^{-1} U^\top \mathcal{R}_s^* U \right) \end{aligned} \quad (62)$$

Proof. The proof revolves around introducing a long-term model, analogous to [7, 31], which can be shown to be accurate for small step-sizes and under conditions 3–4. Details are omitted due to space limitations. We verify the accuracy of numerically in Section 5. \square

We note that for moderately large step-sizes μ , expression (59) will yield more accurate estimates of the steady-state performance than (62). Relation (62) on the other hand is more tractable. We can interpret $U^\top \mathcal{H}^* U$ in (62) as the projection of the Hessian onto the space spanned by U , while $U^\top \mathcal{R}_s^* U$ is the projection of the noise covariance onto the same space. In this sense, $\text{Tr} \left((U^\top \mathcal{H}^* U)^{-1} U^\top \mathcal{R}_s^* U \right)$ is a measure of the inverse signal-to-noise ratio after restricting the signal to the space of feasible solutions in (4). It coincides with the performance of centralized benchmark (44) and the asymptotic performance of [22]. We will verify in Section 5 that (59) is accurate for finite step-sizes μ , and that the proposed bias-corrected algorithm outperforms the approximate solution of [22].

5. NUMERICAL RESULTS

We consider the same setting as [31, Section IV], and refer the reader there for a more detailed motivation of the construction. A total of $K = 50$ agents are placed uniformly in a $[0, 1] \times [0, 1]$ square, and weights $c_{\ell k}$ are assigned between pairs of agents ℓ and k based on their distance (refer to [31, Eq. (83)] for details). Based on these weights, we define the Laplacian matrix $L = \text{diag}\{C \mathbf{1}_M\} - C$, which in turn defines a graph Fourier transform $L = V \Lambda V^\top$. The local models $w_k^o \in \mathbb{R}^5$, which make up $\mathcal{W}^o \in \mathbb{R}^{250}$, are generated by smoothing a randomly generated signal $\mathcal{W} \in \mathbb{R}^{250}$ through a diffusion kernel defined by the graph Fourier transform of L . This results in a collection of local models w_k^o , which vary smoothly over the graph defined by C . Each agent collects linear observations:

$$\gamma_{k,i} = \mathbf{h}_{k,i}^\top w_k^o + \mathbf{v}_{k,i} \quad (63)$$

where $\mathbb{E} \mathbf{h}_{k,i} \mathbf{h}_{k,i}^\top = \sigma_{h,k}^2 I_5$, $\mathbb{E} \mathbf{v}_{k,i}^2 = \sigma_{v,k}^2$, and random variables at different agents are independent. The variances $\sigma_{h,k}^2$ are sampled from the uniform distribution between 0.5 and 2. The local objectives are

$$J_k(w_k) \triangleq \frac{1}{2} \mathbb{E} \left(\gamma_{k,i} - \mathbf{h}_{k,i}^\top w_k \right)^2 \quad (64)$$

and we set V to contain the $P = 3$ leading eigenvectors of U .

We first illustrate the benefit of bias-correction. To this end, we compare the performance of the approximate projection-based algorithm (21) of [22] with the proposed exact subspace diffusion algorithms (32)–(34) for varying noise profiles. In Fig. 1, we sample $\sigma_{v,k}^2$ from the uniform distribution between 0.2 and 0.8 and observe that both (21) and (32)–(34) exhibit similar performance, and match the prediction (59).

We contrast Fig. 1 with a second simulation, where the noise power $\sigma_{v,k}^2$ is now sampled from a the uniform distribution between $0.2 \cdot 10^{-4}$ and $0.8 \cdot 10^{-4}$, and hence much smaller. In Fig. 2 we observe a significant performance advantage in the proposed approach, while the algorithm (21) based on approximate projections shows only mild improvement. This bottleneck is due to the bias induced by employing approximate projections, which remains even as the effect of the gradient noise is reduced, becoming increasingly pronounced the noise vanishes. This is consistent with observations made in the context of consensus optimization [28], where bias-correction yields the significant benefit when employing accurate gradient approximations. In all cases, the performance predictions of Theorem 1 are accurate.

Finally, we illustrate the benefit of allowing for multiple local updates in Fig. 3.

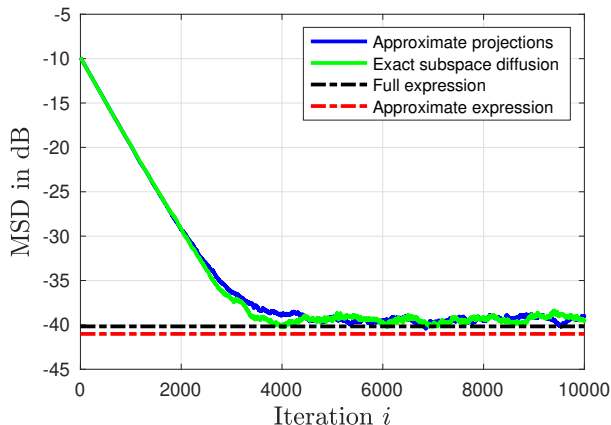


Fig. 1: Comparison in the high-noise setting of the approximate projections-based algorithm (21) and the proposed exact subspace diffusion algorithm (32)–(34) with $E = 1$. We also show the theoretical predictions based on the full expression (59) and approximation (62). In a setting with high noise variance $\sigma_{v,k}^2$, the performance difference is negligible, since the noise dominates any bias induced by (21). Both theoretical expressions match the performance of both algorithms well, with the full expression (59) being more accurate.

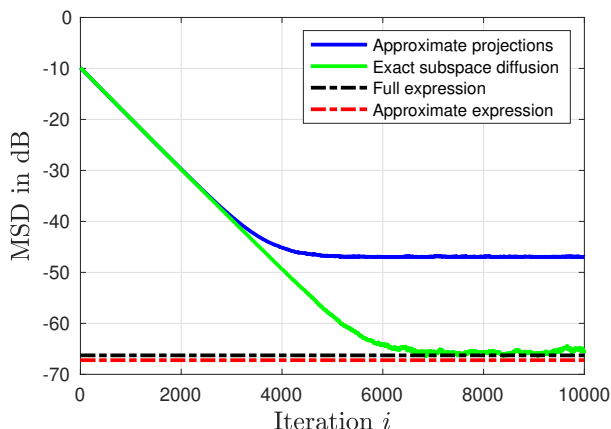


Fig. 2: Comparison in the low-noise setting of the approximate projections-based algorithm (21) and the proposed exact subspace diffusion algorithm (32)–(34) with $E = 1$. We also show the theoretical predictions based on the full expression (59) and approximation (62). In a setting with low noise variance $\sigma_{v,k}^2$, the performance difference is significant, since the bias induced by (21) dominates the noise. Both theoretical expressions match the performance of the proposed algorithm well, with the full expression (59) being more accurate.

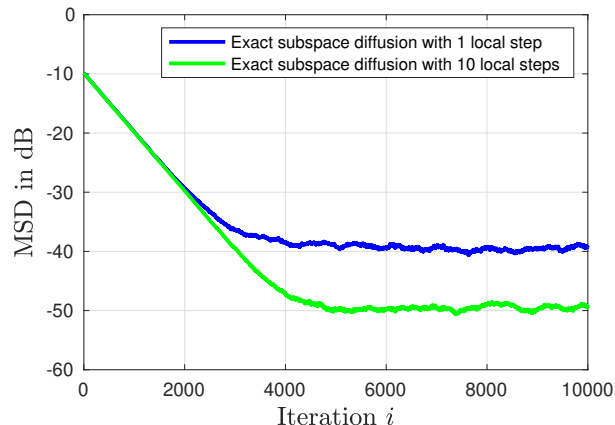


Fig. 3: Comparison of the performance of the exact subspace diffusion algorithm (32)–(34) with $E = 1$ and $E = 10$. Employing 10 local updates results in a 10dB gain in performance, corresponding to 10-fold reduction in steady-state error.

6. CONCLUSION

We have derived exact subspace diffusion, an algorithm for decentralized subspace constrained multitask learning over networks. The construction removes the bias of approximate schemes using primal-dual arguments, and allows for multiple local updates to allow for improved performance without increasing communication load. The algorithm is complemented with a precise characterization of its steady-state error, and simulations illustrating its advantages as well as the accuracy of performance expressions.

7. REFERENCES

- [1] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Trans. Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
- [2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 1997.
- [3] A. Nedić and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan 2009.
- [4] 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.
- [5] P. Kairouz, H. B. McMahan, et al., “Advances and open problems in federated learning,” *Foundations and Trends in Machine Learning*, vol. 14, pp. 1–210, 2021.

- [6] A. H. Sayed, *Inference and Learning from Data*, Cambridge University Press, 2022.
- [7] 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.
- [8] 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.
- [9] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, “Exact diffusion for distributed optimization and learning – Part II: Convergence analysis,” *IEEE Trans. Signal Processing*, vol. 67, no. 3, pp. 724–739, Feb 2019.
- [10] P. Di Lorenzo and G. Scutari, “NEXT: In-network non-convex optimization,” *IEEE Trans. Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.
- [11] A. Mokhtari and A. Ribeiro, “DSA: Decentralized double stochastic averaging gradient algorithm,” *Journal of Machine Learning Research*, vol. 17, no. 61, pp. 1–35, 2016.
- [12] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, “A unified theory of decentralized SGD with changing topology and local updates,” in *Proceedings of the 37th International Conference on Machine Learning*, Jul 2020, vol. 119, pp. 5381–5393.
- [13] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks – Part II: Performance analysis,” *IEEE Trans. Information Theory*, vol. 61, no. 6, pp. 3518–3548, June 2015.
- [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*, 2017, pp. 5336–5346.
- [15] S. Vlaski and A. H. Sayed, “Distributed learning in non-convex environments—Part II: Polynomial escape from saddle-points,” *IEEE Trans. Signal Processing*, vol. 69, pp. 1257–1270, 2021.
- [16] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [17] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning,” in *Advances in Neural Information Processing Systems*, 2017, vol. 30.
- [18] S. Barbarossa, G. Scutari, and T. Battisti, “Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints,” in *Proc. IEEE ICASSP*, 2009, pp. 2893–2896.
- [19] A. Koppel, B. M. Sadler, and A. Ribeiro, “Proximity without consensus in online multiagent optimization,” *IEEE Transactions on Signal Processing*, vol. 65, no. 12, pp. 3062–3077, 2017.
- [20] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, “Multitask learning over graphs: An approach for distributed, streaming machine learning,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14–25, 2020.
- [21] R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed, “Learning over multitask graphs—Part I: Stability analysis,” *IEEE Open Journal of Signal Processing*, vol. 1, pp. 28–45, 2020.
- [22] R. Nassif, S. Vlaski, and A. H. Sayed, “Adaptation and learning over networks under subspace constraints—Part I: Stability analysis,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 1346–1360, 2020.
- [23] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti, “Distributed signal processing and optimization based on in-network subspace projections,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2061–2076, 2020.
- [24] M. Kayaalp, S. Vlaski, and A. H. Sayed, “Dif-MAML: Decentralized multi-agent meta-learning,” *IEEE Open Journal of Signal Processing*, vol. 3, pp. 71–93, 2022.
- [25] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, “Graph signal processing for machine learning: A review and new perspectives,” *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 117–127, 2020.
- [26] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks - Part I: Transient analysis,” *IEEE Trans. Information Theory*, vol. 61, no. 6, pp. 3487–3517, June 2015.
- [27] 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.
- [28] K. Yuan, S. A. Alghunaim, B. Ying, and A. H. Sayed, “On the influence of bias-correction on distributed stochastic optimization,” *IEEE Trans. Signal Processing*, vol. 68, pp. 4352–4367, 2020.
- [29] M. Marquis, D. Abou Jaoude, M. Farhood, and C. A. Woolsey, “Robustness and convergence analysis of first-order distributed optimization algorithms over subspace constraints,” available as *arXiv:2210.16277*, Oct 2022.
- [30] S. Vlaski, S. Kar, A. H. Sayed, and J. M. F. Moura, “Networked signal and information processing,” available as *arXiv:2210.13767*, Oct 2022.
- [31] R. Nassif, S. Vlaski, and A. H. Sayed, “Adaptation and learning over networks under subspace constraints—Part II: Performance analysis,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2948–2962, 2020.
- [32] S. Vlaski and A. H. Sayed, “Second-order guarantees of stochastic gradient descent in nonconvex optimization,” *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6489–6504, 2022.