Distributed Personalized Gradient Tracking with Convex Parametric Models

Distributed Personalized Gradient Tracking With Convex Parametric Models

in:

IEEE Transactions on Automatic Control, vol. 68, no. 1, pp. 588-595

The final published version is available online at:
https://doi.org/10.1109/TAC.2022.3147007

Distributed Personalized Gradient Tracking with Convex Parametric Models

Ivano Notarnicola *Member, IEEE*, Andrea Simonetto *Member, IEEE*, Francesco Farina *Member, IEEE*, Giuseppe Notarstefano *Member, IEEE*

***Abstract*—We present a distributed optimization algorithm for solving online personalized optimization problems over a network of computing and communicating nodes, each of which linked to a specific user. The local objective functions are assumed to have a composite structure and to consist of a known time-varying (engineering) part and an unknown (user-specific) part. Regarding the unknown part, it is assumed to have a known parametric (e.g., quadratic) structure a priori, whose parameters are to be learned *along with* the evolution of the algorithm. The algorithm is composed of two intertwined components: (i) a dynamic gradient tracking scheme for finding local solution estimates and (ii) a recursive least squares scheme for estimating the unknown parameters via user's noisy feedback on the local solution estimates. The algorithm is shown to exhibit a bounded regret under suitable assumptions. Finally, a numerical example corroborates the theoretical analysis.**

***Index Terms*—Distributed Optimization, Distributed Learning, Online Optimization.**

I. Introduction

Cyber-physical and social systems (CPSS) are becoming increasingly important in today's society, whenever human actions, preferences, and behaviors are added to the cyber and physical space [1]. Important examples of this class of systems are the energy grid [2], [3], transportation infrastructures [4], personalized healthcare [5], and robotics [6].

A key feature of CPSS is the trade-off between given engineered performance metrics and user's (dis)comfort, perceived safety, and preferences. While, on one side, engineered goals may come from well-defined metrics based on physical models and can be time-varying to model data streams [7], on the other side, user's (dis)satisfaction is more complex to model. The "utility" function to be optimized for the users is often based on averaged models constructed on generic one-fits-all models. However, good averaged models of users' utilities are difficult to obtain for the associated cost and time of human studies, the data is therefore scarce and biased. For these reasons, more tailored and personalized strategies are to be preferred when dealing with humans [8].

This paper studies time-varying optimization problems distributed across a network of N agents. Each agent represents both a physical node (e.g., a home or a car) and its associated user. The optimization has a cost function that comprises of both a known time-varying engineering cost, and an unknown user specific (dis)satisfaction function. Formally, we define the *distributed personalized problem* as

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \sum_{i=1}^{N} \underbrace{V_i(x;t)+U_i(x)}_{f_i(x;t)}, \quad t\in\mathbb{N} \tag{1}$$

where $x\in\mathbb{R}^n$ represents the common decision variable, and $t\in\mathbb{N}$ represents the time index; each agent i is equipped with the known time-varying engineering cost $V_i(x;t):\mathbb{R}^n\times\mathbb{N}\to\mathbb{R}$ and with the unknown user's dissatisfaction function $U_i(x):\mathbb{R}^n\to\mathbb{R}$. The aggregated cost $f_i(x,t):\mathbb{R}^n\times\mathbb{N}\to\mathbb{R}$ is associated to agent i only. Then, by *solving* problem (1), we mean to generate a sequence of tentative solutions, say $\{\bar{x}_t\}_{t=1}^T$, which make the corresponding cost $\sum_{i=1}^N f_{i,t}(\bar{x}_t;t)$ as close as possible to its (current) optimal value, say $f_\star(t)$, for all t. In particular, as customary in online optimization, we measure the quality of the given sequence $\{\bar{x}_t\}_{t=1}^T$ using the *cumulative dynamic regret* up to time T defined by

$$R_T(\{\bar{x}_t\}_{t=1}^T) \triangleq \sum_{t=1}^{T}\left(\sum_{i=1}^{N} f_i(\bar{x}_t;t)-f_\star(t)\right) \tag{2}$$

and the *average dynamic regret* up to time T defined by $R_T(\{\bar{x}_t\}_{t=1}^T)/T$. As it is customary in the distributed setting, we also complement these measures with the consensus metric $C_T(\{x_{i,T}\}_{i=1}^N,\bar{x}_T)\triangleq\sum_{i=1}^N\|x_{i,T}-\bar{x}_T\|^2$, quantifying how far from consensus the local decisions $x_{i,T}$ are at time T.

The challenges in solving problem (1) are multiple. First, a *distributed* strategy must be developed. Then, not only the optimization problem changes over time, but its cost function is not completely known by the agents and it has to be learned *concurrently* to the solution of the problem, by employing noisy user's feedback.

This paper addresses all the above mentioned challenges and provides the following main contributions.

We propose a *personalized gradient tracking* distributed scheme combining an online optimization algorithm with a learning mechanism, and derive a bound on its dynamic regret. As a building block for the proposed scheme, we develop a dynamic gradient tracking algorithm that, given a smooth strongly convex time-varying cost function, is capable of tracking its solution sequence $\{x_\star(t)\}_{t\in\mathbb{N}}$ in a distributed way up to a bounded error, in line with time-varying optimization results [7], [9]. Notice that, this block is a contribution per se to the distributed online optimization literature.

I. Notarnicola, F. Farina and G. Notarstefano are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy, name.lastname@unibo.it.

A. Simonetto is now with the Department of Applied Mathematics, ENSTA-Paris, Palaiseau, France. He performed part of this work while at IBM Research Europe, Dublin, Ireland, andrea.simonetto@ensta-paris.fr.

This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART).

In the proposed personalized gradient tracking strategy, the dynamic gradient tracking update is interlaced with a learning mechanism to let each node learn the user's cost function $U_i(x)$, by employing noisy user's feedback in the form of a scalar quantity given by $y_{i,t} = U_i(x_{i,t}) + \epsilon_{i,t}$, where $x_{i,t}$ is the local, tentative solution at time t and $\epsilon_{i,t}$ is a noise term. It is worth pointing out that in this paper, we consider convex parametric models, instead of more generic non-parametric models, such as Gaussian Processes [3], [8], [10], [11], or convex regression [12], [13]. The reasons for this choice stem from the fact that *(i)* user's functions are or can be often approximated as convex (see, e.g., [14], [15] and references therein), which makes the overall optimization problem much easier to be solved; *(ii)* convex parametric models have better *asymptotical rate* bounds[1] than convex non-parametric models [12], which is fundamental when attempting at learning with scarce data; and *(iii)* a solid online theory already exists in the form of recursive least squares (RLS) [16]–[18]. Therefore, our learning mechanism is based on a RLS algorithm, whose asymptotical rate is characterized.

Although the high-level algorithmic idea of combining a distributed (online) optimization update with a recursive regression scheme appears intuitively reasonable, the concurrent application of the two updates at the same time scale introduces *several challenges* in the analysis that have been addressed by properly applying and adapting tools from online and distributed optimization and from parameter estimation.

To summarize, the main goal of the paper is to provide a first-of-its-kind algorithm to simultaneously learn and solve optimization problems with unknown convex parametric models online and in a distributed fashion, while at the same time incorporating human preferences in the loop.

Literature survey: A centralized bandit framework with a similar structure to the one considered in this paper has been introduced in [8], even though in the context of non-parametric learning (see also references therein for a comprehensive literature survey).

In the distributed setup addressed by this paper, we assume that the function U_i can be modeled as a linearly parametrized convex quadratic function, whose parameters are unknown and have to be learned. This represents a first step towards generic parametric models[2]. Non-parametric approaches in the literature to learn unknown functions are e.g., (shape-constrained) Gaussian processes [3], [10] and convex regression [12], [13]. As said, we prefer here parametric models for their faster asymptotical rates, cheap online computational load, and ease of introducing convexity constraints.

Another line of research, not followed in this paper, is zero-order (stochastic) online convex optimization, where the cost function is assumed convex, but not known, and its gradient is estimated by function evaluations [20], [21]. Even though this line of research is extremely relevant for human-in-the-loop settings (see, e.g., [6]), we distinguish ourselves from it since we do not assume that the user's feedback is available at each time t. This is key in human systems where feedback may come intermittently, and still one needs to be able to solve the optimization problem. Imagine for example that a particular user is content with whichever decision and she/he does not feel the need for giving feedback, after a few initial ones. Then our algorithm would work seamlessly, since it builds a model for U_i, while zero-order methods would still need function evaluations (i.e., feedback) to proceed.

Regarding optimization problems with (known) time-varying cost function, they have been addressed in the distributed optimization literature, both in the stochastic (see, e.g., [22], [23] and references therein) and online/time-varying settings, e.g., [7], [24]–[29], and references therein. Our algorithm relies on the so-called gradient tracking algorithm firstly proposed in [30]–[33]). The gradient tracking scheme has been originally designed for static optimization problems while it has been applied later to online problems in, e.g., [29], [34]. The most important difference here is that not knowing either the cost function, the minimum dynamics, or both, poses *important additional challenges in ensuring convergence concurrently with learning*.

Notation: The j-th component of a vector v is $[v]_j$ while the j-th row of a matrix A is $[A]_j$. For m vectors $v_1, \dots, v_m$, we define $\mathrm{col}(v_1, \dots, v_m) \triangleq [v_1^\top, \dots, v_m^\top]^\top$. Given $c \in \mathbb{R}$, $b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$, let $v \triangleq \mathrm{col}(c, b, [A]_1^\top, \dots, [A]_n^\top) \in \mathbb{R}^{1+n+n^2}$, then we define the operator $\textsc{unpack}(v)$ so that $(A, b, c) = \textsc{unpack}(v)$. The all-one vectors of appropriate dimension is $\mathbf{1}$. Gradients w.r.t. the variable x of the function $f(x;t)$ are indicated with $\nabla f(x;t)$.

II. Problem Assumptions

Problem (1) is to be solved in a distributed way by a network of N agents. We have depicted the problem setting in Figure 1: each agent is composed by a physical node (e.g., a home, a car, a mobile phone) linked to an end-user. The nodes are equipped with a time-varying cost V_i and can evaluate a noisy version of U_i by asking the user for feedback on a particular decision $x_{i,t}$. Each node can compute and communicate with its direct neighbors over a fixed network. In this context, each agent i has only a *partial* knowledge of the target problem.

We consider the following assumption on the problem structure.

Assumption II.1. *For all $i = 1 \dots, N$ it holds that:*

(i) The function $V_i(x;t)$ is m_V-strongly convex and its gradients are L_V-Lipschitz continuous for all $t \in \mathbb{N}$.

(ii) The function $U_i(x)$ has a quadratic structure, i.e., $U_i(x) = \frac{1}{2} x^\top P_i x + q_i^\top x + r_i$, with $P_i \in \mathbb{R}^{n \times n}$ symmetric

[1] By asymptotical rate, we mean how the approximation gets closer to the true function as the number of data points (feedback) increases. Shape-constrained Gaussian processes can be used to impose convexity constraints in a practical sense, but their computational complexity scales as $O(t^3)$, where t is the number of data points, they are not trivially extended for decision spaces with dimensions $n > 1$, and asymptotical rate bounds are not yet available. Convex regression has asymptotical rate bounds of the form of $O(t^{-1/n})$, which is very slow compared to the parametric models, and their computational complexity scales at least as $O(t^2 n^3)$.

[2] The approach in this paper can be extended to linearly parametrized convex functions, but we assume a quadratic structure for the sake of clarity. If the user's parametric model is more complex, we can always focus on local results, where the model is approximately convex and linear in the parameters, see also [19] for examples of linearly parametrized models applied to inverse control and optimization, which are close in spirit to our problem.

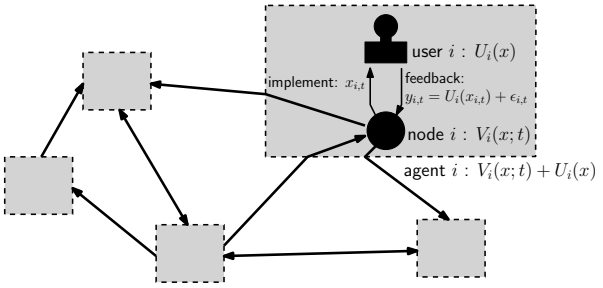

Fig. 1. The problem setup: a network of connected and communicating nodes, each node with associated an end-user from which feedback may be asked on their own dissatisfaction on a particular decision.

and with eigenvalues in the range $[m_i, L_i]$*, with* $L_i \geq m_i > 0$*,* $q_i \in \mathbb{R}^n$*,* $r_i \in \mathbb{R}$*.*

(iii) The parameters P_i*,* q_i *and* r_i *of* U_i *are* unknown*, however one knows a (loose) bound on* L_i*, and noisy measurements of* $U_i(x)$ *can be taken for any point* $x \in \mathbb{R}^n$ *as* $y_i = U_i(x) + \epsilon_i$*,where* ϵ_i *denotes a generic scalar zero-mean noise with finite variance.*

(iv) The optimizer of problem (1)*,* $x_\star(t)$*, is finite for each* $t \in \mathbb{N}$*, and* $\|x_\star(t)\| < \infty$*.* □

Assumption II.1 on the engineering function $V_i(x;t)$ is quite standard in the time-varying literature [7], [9].

As for the the m_i-strongly convex, L_i-smooth quadratic model of $U_i(x)$, we point out that, though partially restrictive, this structure is reasonable as discussed in the introduction (see also Footnote 2) and it can be relaxed. Loose bounds on L_i can be obtained from experiments and average user data. Finally, the finiteness assumption on the optimizer (which exists and it is unique for *(i)-(ii)*) just ensures that the problem is well-posed even in a time-varying setting.

Since each $U_i(x)$ is quadratic but unknown, its parameters need to be estimated over time. Therefore, we let each agent i consider an approximation of $U_i(x)$ at each time t given by

$$\hat{U}_{i,t}(x) \triangleq \tfrac{1}{2}\, x^\top \hat{P}_{i,t} x + \hat{q}_{i,t}^\top x + \hat{r}_{i,t}, \tag{3}$$

where $\hat{P}_{i,t}, \hat{q}_{i,t}$ and $\hat{r}_{i,t}$ represent the current estimates of the true (unknown) parameters P_i, q_i and r_i. We then define the local estimated cost of agent i as

$$\hat{f}_i(x;t) \triangleq V_i(x;t) + \hat{U}_{i,t}(x).$$

Defining $\hat{f}(x;t) \triangleq \sum_{i=1}^N \hat{f}_i(x;t)$, we denote by $\hat{f}_\star(t)$ its minimum value and by $\hat{x}_\star(t)$ the minimizer. Consistently, we define $f(x;t) \triangleq \sum_{i=1}^N f_i(x;t)$ and its minimum value $f_\star(t)$ attained at some $x_\star(t)$.

At this point, we make no specific choice on the type of estimation/learning algorithm to determine $\hat{U}_{i,t}(x)$, provided that it satisfies the following.

Assumption II.2. *For the chosen estimation algorithm, the estimated* $\hat{U}_{i,t}(x)$ *is bounded for any finite* x*, for all* i *and* t*. Moreover:*

(i) With high probability*, the estimated* $\hat{P}_{i,t}$ *is symmetric and it has eigenvalues in the set* $[0, \mu L_i]$*,* $\mu > 1$*. I.e., for any* $\delta \in (0,1]$ *and* $\mu > 1$*, there exists a finite* $\bar{t}$*, for which:*

$$\mathbf{Pr}(\mu L_i I_n \geq \hat{P}_{i,t} \geq 0 \,|\, \forall t \geq \bar{t}) \geq 1 - \delta,$$

(ii) When the first fact holds true, there exist constants $c_x, c_\nabla < \infty$ *such that:*

$$\|\hat{x}_\star(t) - \hat{x}_\star(t-1)\| \leq c_x,$$
$$\max_i \|\nabla \hat{f}_i(\hat{x}_\star(t);t) - \nabla \hat{f}_i(\hat{x}_\star(t);t-1)\| \leq c_\nabla. \qquad \square$$

Assumption II.2(i) is a mild assumption, and it will hold for our RLS scheme [Cf. Appendix C]. It imposes that eventually (and with high probability), the estimated values of $\hat{P}_{i,t}$ get close to obtain the properties of the true P_i.

Once $\mu L_i I_n \geq \hat{P}_{i,t} \geq 0$, then the approximate problems are convex and for Assumption II.1, the optimizer of $\hat{f}(x;t)$ is finite. Then, Assumption II.2(ii) is mild and standard in time-varying optimization: it ensures that the problem changes are bounded. This in turn guarantees that one is able to track its solution up to a meaningful error bound.

Remark II.3. *A key aspect in time-varying optimization is the* $O(T)$ *path length, defined as* $P_T = \sum_{t=1}^T \|\hat{x}_\star(t) - \hat{x}_\star(t-1)\|$*. This is different from (bandit) online convex optimization which often assumes* $P_T = o(T)$ *or finite* P_T*. For a* $O(T)$ *path length, one cannot expect less than finite asymptotic error bounds and* $O(T)$ *cumulative dynamic regret bounds [7], [25], [35]. The reader can verify that when* $P_T = o(T)$*, then* c_x, c_∇ *must be functions of time, and they need to vanish as time increases. In such a setting one can show* $o(T)$ *dynamic regret.* □

With Assumption II.2 in place, after $\bar{t}$ and for all $t \geq \bar{t}$, the approximate cost function $\hat{f}(x;t)$ is m-strongly convex and L-smooth with $m = Nm_V$, $L = NL_V + \mu \sum_{i=1}^N L_i$, with probability $1-\delta$, and the local cost function $\hat{f}_i(x;t)$ is $(L_V + \mu L_i)$-smooth. In addition, and with Assumption II.1(ii), for the gradient $\nabla \hat{f}_i(x;t)$ one has that

$$\begin{aligned}\|\nabla \hat{f}_i(x;t) - \nabla \hat{f}_i(x;t-1)\| &= \|\nabla \hat{f}_i(x;t) - \nabla \hat{f}_i(x;t-1) \\ &\quad \pm (\nabla \hat{f}_i(\hat{x}_\star(t);t) - \nabla \hat{f}_i(\hat{x}_\star(t);t-1))\| \\ &\leq 2(L_V + \mu L_i)\|x - \hat{x}_\star(t)\| + c_\nabla, \end{aligned} \tag{4}$$

with probability $1-\delta$. In addition, the estimation error $|\hat{U}_{i,t}(x) - U_i(x)|$ is bounded for any finite x since $\hat{U}_{i,t}(x)$ is proper, and one can define the *estimation error length* as,

$$c_U := \sum_{t=1}^T |\hat{U}_{i,t}(x) - U_i(x)|. \tag{5}$$

Under the reasonable assumption that the estimator delivers a bounded error estimation, i.e., $|\hat{U}_{i,t}(x) - U_i(x)| < c_u < \infty$ for all $i, x, t \geq \bar{t}$, then $c_U = O(T)$. More sensible estimation algorithms will yield $c_U = o(T)$, as we will show.

Regarding the structure of the communication network, it is modeled through a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ in which $\mathcal{V} = \{1, \ldots, N\}$ denotes the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges and $\mathcal{W} = [w_{ij}] \in \mathbb{R}^{N \times N}$ the weighted adjacency matrix. We let $\mathcal{G}$ satisfy the following.

Assumption II.4. *The graph* $\mathcal{G}$ *is directed and strongly connected. The weighted adjacency matrix* $\mathcal{W}$ *is doubly-stochastic, i.e.,* $\sum_{j=1}^N w_{ij} = 1$ *for all* $i = 1, \ldots, N$ *and* $\sum_{i=1}^N w_{ij} = 1$ *for all* $j = 1, \ldots, N$*. Moreover, for all* $i = 1, \ldots, N$*,* $w_{ij} > 0$ *if and only if* $j \in N_i$*, where* $N_i \triangleq \{j \mid (j,i) \in \mathcal{E}\} \cup \{i\}$ *is the set of in-neighbors of node* i*.* □

The condition above does not include all possible communication topologies, however it includes the broad class of balanced digraphs. See [36] for further details.

III. Personalized Gradient Tracking Distributed Algorithm

We describe now our novel distributed online algorithm for solving Problem (1), along with its theoretical properties.

A. Distributed Algorithm Description

Each agent i stores and updates several states. First, it has a local estimate $x_{i,t} \in \mathbb{R}^n$ of the solution of problem (1) at iteration t. Second, it maintains local estimates $\hat{P}_{i,t} \in \mathbb{R}^{n\times n}$, $\hat{q}_{i,t} \in \mathbb{R}^n$ and $\hat{r}_{i,t} \in \mathbb{R}$ of the unknown parameters of the local function $U_i(x)$ (cf. (3)). Third, it uses an auxiliary state $d_{i,t} \in \mathbb{R}^n$ to reconstruct an the current value of the gradient of $\sum_{i=1}^N \hat{f}_i(x_{i,t};t)$.

For computational convenience, the local variable $x_{i,t}$ will be often arranged in the following vectorized form

$$\boldsymbol{\chi}_{i,t} = \mathrm{col}(1, x_{i,t}, [x_{i,t}]_1 x_{i,t}/2, \dots, [x_{i,t}]_n x_{i,t}/2) \in \mathbb{R}^{1+n+n^2}.$$

Each iteration $t \in \mathbb{N}$ of the distributed algorithm consists in three consecutive actions performed by each agent i.

1) A *feedback* on the current local solution estimate $x_{i,t}$ is obtained from the user. In particular, a noisy *measurement* of the output of $U_i(\cdot)$ evaluated at $x_{i,t}$ is computed and stored as $y_{i,t}$ given in (6).
2) The estimates $\hat{P}_{i,t}$, $\hat{q}_{i,t}$ and $\hat{r}_{i,t}$ of the unknown parameters P_i, q_i and r_i of U_i are updated by means of an ad-hoc *learning* procedure (7). This procedure relies on a RLS scheme which makes use only of the most updated data $(y_{i,t}, x_{i,t})$, thus not requiring to store and use all the past points generated by the distributed algorithm.
3) The local solution estimate $x_{i,t}$ of problem (1) at time t is updated via a *dynamic gradient tracking* distributed algorithm (9), whose aim is to track the sequence of solutions $\{x_\star(t)\}_{t\in\mathbb{N}}$ of problem (1).

Algorithm 1 reports the pseudocode of the proposed scheme, with step-size $\alpha > 0$ and tuning parameter $\eta \gg 0$.

Remark III.1. *We assume that the users give feedback at each time t that they are asked for it, with no delay. This is not a limitation: we could consider cases in which users give intermittent feedback at different time-scales and with delays. This would mean that the learning phrase described by* (6)-(7)-(8) *would be slower than the optimization process* (9)*. From the optimization perspective, since the knowledge of U_i changes every time a new feedback is received, the worst case scenario is when feedback is given at each time t (see also [8]).* □

B. Parameters Estimation via Recursive Least Squares (RLS)

The aim of the learning part of Algorithm 1 (cf. (7)) is to provide a recursive scheme to let each agent i estimate the unknown parameters of U_i. Specifically, the considered scheme aims at solving, for each t, the least squares (LS) problem

$$\underset{P\in\mathbb{R}^{n\times n},\, q\in\mathbb{R}^n,\, r\in\mathbb{R}}{\text{minimize}} \ \sum_{s=1}^{t} \left(\tfrac{1}{2}\, x_{i,s}^\top P x_{i,s} + q^\top x_{i,s} + r - y_{i,s}\right)^2, \tag{10}$$

for a given set of estimate-measurement pairs $(x_{i,s}, y_{i,s})_{s=1}^t$.

By defining $\xi_i \triangleq \mathrm{col}(r, q, [P]_1^\top, \dots, [P]_n^\top) \in \mathbb{R}^{1+n+n^2}$, problem (10) can be equivalently recast into

$$\hat{\xi}_{i,t} = \arg\min_{\xi_i} \ \sum_{s=1}^{t} (\xi_i^\top \boldsymbol{\chi}_{i,s} - y_{i,s})^2, \tag{11}$$

and $\hat{P}_{i,t}, \hat{q}_{i,t}$ and $\hat{r}_{i,t}$ can be then retrieved from $\hat{\xi}_{i,t}$ via (8) (cf. the Notation) and then made symmetric. Now, instead of keeping track of all the data, problem (11) is solved as data become available by means of a RLS approach [16, Chap. 11], yielding (7) in Algorithm 1.

The estimate computed by using RLS differs from the standard, non recursive, least squares (LS) counterpart only in the initial iterations, due to the initialization, which is quickly negligible [16, Chap. 11]; the asymptotic convergence properties coincide with those of the non recursive LS approach. Upon defining $\xi_{i,\star} = \mathrm{col}(r_i, q_i, [P_i]_1^\top, \dots, [P_i]_n^\top)$ for all i, then for each agent the following classical result holds.

Lemma III.2 (Large sample aymptotic properties of LS)**.** *Let the data sequence $\{(\boldsymbol{\chi}_{i,s}, y_{i,s})\}_{s\geq 0}$ be such that:*

Algorithm 1 Personalized Gradient Tracking

Initialization: $x_{i,0}$ arbitrary, $d_{i,0} = \nabla \hat{f}_i(x_{i,0};0)$, $R_{i,0} = \eta I_{1+n+n^2}$, $\hat{\xi}_{i,0} = 0$.

Evolution: $t = 1, 2, \dots$

MEASURING/FEEDBACK

$$y_{i,t} = U_i(x_{i,t-1}) + \epsilon_{i,t} \tag{6}$$

LEARNING

$$\mathbf{s}_{i,t} = \frac{R_{i,t-1}\boldsymbol{\chi}_{i,t}}{1 + \boldsymbol{\chi}_{i,t}^\top R_{i,t-1}\boldsymbol{\chi}_{i,t}} \tag{7a}$$

$$R_{i,t} = R_{i,t-1} - (1 + \boldsymbol{\chi}_{i,t}^\top R_{i,t-1}\boldsymbol{\chi}_{i,t})\mathbf{s}_{i,t}\mathbf{s}_{i,t}^\top \tag{7b}$$

$$\hat{\xi}_{i,t} = \hat{\xi}_{i,t-1} + (y_{i,t} - \boldsymbol{\chi}_{i,t}^\top \hat{\xi}_{i,t-1})\mathbf{s}_{i,t} \tag{7c}$$

$$(\hat{P}_{i,t}, \hat{q}_{i,t}, \hat{r}_{i,t}) = \textsc{unpack}(\hat{\xi}_{i,t}), \quad \hat{P}_{i,t} \leftarrow (\hat{P}_{i,t} + \hat{P}_{i,t}^\top)/2 \tag{8}$$

DYNAMIC GRADIENT TRACKING

$$x_{i,t} = \sum_{j\in\mathcal{N}_i} w_{ij} x_{j,t-1} - \alpha\, d_{i,t-1} \tag{9a}$$

$$g_{i,t} = \nabla V_i(x_{i,t};t) + \hat{P}_{i,t}^\top x_{i,t} + \hat{q}_{i,t} \tag{9b}$$

$$d_{i,t} = \sum_{j\in\mathcal{N}_i} w_{ij} d_{j,t-1} + (g_{i,t} - g_{i,t-1}) \tag{9c}$$

- *the $\{(\boldsymbol{\chi}_{i,s}, y_{i,s})\}_{s\geq 0}$ is a realization of a jointly stationary and ergodic stochastic process;*
- *the matrix $\Sigma_{xx} = \mathbb{E}[\boldsymbol{\chi}_{i,s}\boldsymbol{\chi}_{i,s}^\top]$ is nonsingular;*
- *for $\omega_{i,s} \triangleq \boldsymbol{\chi}_{i,s}\epsilon_{i,s}$, then $\{\omega_{i,s}\}$ is a martingale difference sequence with finite second moments (cfr. [17, Assumption 2.5]), and denote $S = \mathbb{E}[\omega_{i,s}\omega_{i,s}^\top]$.*

Then,

$$\sqrt{t}(\hat{\xi}_{i,t} - \xi_{i,\star}) \xrightarrow{D} \mathcal{N}(0, \Sigma_{xx}^{-1} S \Sigma_{xx}^{-1}), \text{ as } t \to \infty, \qquad (12)$$

where the notation $\xrightarrow{D}$ stands for convergence in distribution.

Proof. See, e.g., [17, Prop. 2.1] and [16, Chap. 8, 9, 11]. □

Result (12) implies that the random variable $\sqrt{t}(\hat{\xi}_{i,t} - \xi_{i,\star})$ is asymptotically normal distributed, and that $\|\hat{\xi}_{i,t} - \xi_{i,\star}\| \to 0$ with rate $O(1/\sqrt{t})$. That is, the rate $O(1/\sqrt{t})$ is the asymptotical rate bound for (R)LS, and this will help us show that the estimation length $c_U = O(\sqrt{T})$.

The assumptions in Lemma III.2 require some words when applied to our setting. Since the regressors $\boldsymbol{\chi}_{i,s}$ are determined by the gradient tracking process, and ultimately (upon convergence) they are close to the optimizer trajectory, we are requiring that the optimizers $\{\boldsymbol{x}_\star(t)\}$: (i) eventually behave as a stationary and ergodic process, and (ii) are never exactly the same (so that Σ_{xx} remains non-singular). In practice in our model the optimizers change in time due to external, time-varying data-streams (which could be assumed stationary and ergodic) and, thus, satisfy this assumption.

C. Dynamic Gradient Tracking

The step in (9) is meant to implement a gradient tracking distributed algorithm tailored for an online optimization problem, whose convergence is provided next.

Theorem III.3. *Consider the sequence $\{x_{i,t}\}_{t\geq 1}$ generated by* (9) *and let $\bar{x}_t \triangleq \frac{1}{N}\sum_{i=1}^N x_{i,t}$. Let Assumptions II.1, II.2, and II.4 hold. Choose a $\mu > 1$. Then, there exist a $\rho < 1$ and a small enough step-size α in $(0, N/L]$, for which the following holds with high probability*

$$\limsup_{t\to\infty} \sum_{i=1}^N \hat{f}_i(\bar{x}_t; t) - \hat{f}_\star(t) = \frac{L(Nc_\nabla^2 + c_x^2)}{2(1-\rho)^2} =: \frac{L}{2}\bar{c}^2$$

with linear rate ρ. The consensus metric C_T satisfies $\limsup_{T\to\infty} C_T(\{x_{i,T}\}_{i=1}^N, \bar{x}_T) = \bar{c}^2$, and the average $\bar{x}_T$ is bounded. □

The proof of Theorem III.3 is given in Appendix B. The result is in line with current works in time-varying optimization [7], [9], as well as regret results with dynamic comparators when the path length grows as $O(T)$ and we employ a constant step-size [25].

D. Regret Analysis of Algorithm 1

The next theorem, whose proof is reported in Appendix D, represents the second main result of this paper. It shows that a bound on the cumulative regret can be provided under suitable assumptions, and that the asymptotic average regret is bounded.

Theorem III.4. *Let the sequences $\{(\boldsymbol{\chi}_{i,t}, y_{i,t})\}_t$ be generated by Algorithm 1. Let Assumptions II.1, II.2(ii) and II.4 hold. Choose a $\mu > 1$. Then, there exist a $\rho < 1$ and a small enough step-size α in $(0, N/L]$, for which w.h.p.*

$$R_T(\{\bar{x}_t\}_{t=1}^T) \leq O(1) + O(c_U) + O\left(T\,\frac{L(Nc_\nabla^2 + c_x^2)}{2(1-\rho)^2}\right).$$

Moreover, w.h.p., the average dynamic regret reaches an asymptotical value as

$$\limsup_{T\to\infty} \frac{R_T(\{\bar{x}_t\}_{t=1}^T)}{T} = O\left(\frac{c_U}{T}\right) + \frac{L}{2}\bar{c}^2 = O(1).$$

Finally, w.h.p., the consensus metric C_T is such that $\limsup_{T\to\infty} C_T(\{x_{i,T}\}_{i=1}^N, \bar{x}_T) = \bar{c}^2$. □

Algorithm 1 delivers a bounded average dynamic regret with high probability. In particular, the dynamic regret is composed of three terms. The first $O(1)$ term collects the initialization errors (e.g., when $\hat{U}_{i,t}$ is nonconvex). The second $O(c_U)$ term, more standard, represents the learning bound. (It is in general $O(T)$, but $O(\sqrt{T})$ if the assumptions of Lemma III.2 are verified, see Appendix C (Lemma A.5), thereby vanishing as $O(1/\sqrt{T})$ in the average regret result). Finally, the third $O(T)$ term pertains the tracking of the distributed solution trajectory, and it is linear in T since the path length is linear in T [25]. The asymptotical bound depends on how fast the problems are changing in time, due to variations of the gradients and the optimizers, as typical in time-varying optimization. Finally, note that Assumption II.2(i) is not required here, since it is verified for our RLS scheme [Cf. Appendix C].

Remark III.5 (Regret in a distributed setting)**.** *Under boundedness of the consensus metric C_T given by Theorem III.4, an agent j-specific regret bound $\sum_{t=1}^T \sum_{i=1}^N f_i(x_{j,t}; t) - f_\star(t)$ can also be derived, with the same convergence rate, and leading term of $O(c_U) + O(2T\bar{c}^2)$ (Cf. [37, Appendix **??**]).*□

E. Computational and communication complexity

We finish our analysis of Algorithm 1 by reporting its computational and communication complexity. First, only local computations are carried out, and the most demanding are matrix/vector multiplications on vector $\boldsymbol{\chi}_{i,t} \in \mathbb{R}^{1+n+n^2}$, delivering a computational complexity of $O(n^4)$. This is in comparison with Gaussian Processes $O(t^3)$ and convex regression $O(t^2n^3)$ [Cf. Footnote 1]. This makes our method less computational intensive than other techniques, especially for large $t \gg n$ (i.e., as more and more data comes in). This is due to the fact that our method is recursive.

As for the communication complexity, our gradient tracking employs two communication rounds for each iteration for a total of at worst $4(N-1)n$ scalar sent.

IV. Numerical Example

We consider a scenario with both V_i and U_i quadratic, i.e.,

$$\underset{x\in\mathbb{R}^3}{\text{minimize}} \sum_{i=1}^N \Big(\underbrace{\|x - p_i(t)\|^2}_{V_i(x;t)} + \underbrace{\|x - v_i\|^2}_{U_i(x)} \Big), \qquad t \geq 0.$$

with $p_i(t) \in \mathbb{R}^3$ for all t and $v_i \in \mathbb{R}^3$.

We implemented the Personalized Gradient Tracking Algorithm 1 with DISROPT [38] and performed a simulation with $N = 30$ agents, in which each target speed $p_i(t)$ evolves according to the following law

$$p_i(t) = z_i + \psi_i \sin(t/m_i)$$

with $z_i \in \mathbb{R}^3$, $\psi_i \in \mathbb{R}^3$ and $m_i > 1$. We randomly generate the coefficients by picking $v_i \in \mathcal{U}[-1.5, 1.5]^3$, $z_i \in \mathcal{U}[-5, 5]^3$, $m_i \in \mathcal{U}[100, 150] \cap \mathbb{N}$, $\psi_i \in \mathcal{U}[0.5, 0.6]$ and $\epsilon_{i,t} \in \mathcal{N}(0, 0.2)$ for all $i = 1, \ldots, N$. We ran 10^6 iterations with step-size $\alpha = 0.01$ and initial conditions $x_{i,0} \in \mathcal{U}[-1.5, 1.5]^2$. The evolution of the average regret R_t/t obtained by Algorithm 1 is shown in Figure 2. Specifically, we evaluate the dynamic regret as expressed in (2) at $\bar{x}_t \triangleq \frac{1}{N}\sum_{i=1}^N x_{i,t}$ for all $t = 1, \ldots, 10^6$. As expected from Theorem III.4, the average regret decays to some constant value.

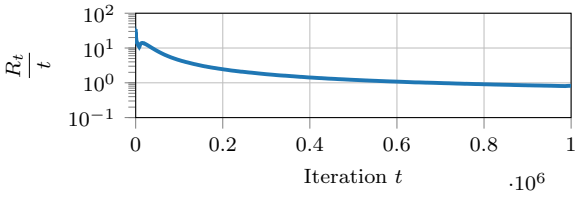

Fig. 2. Evolution of the average regret.

Figure 3 shows the consensus and tracking error. In particular, it can be appreciated that they become stationary, though not vanishing, after the initial transient highlighted in the insets, consistently with the theoretical bound proved by (18).

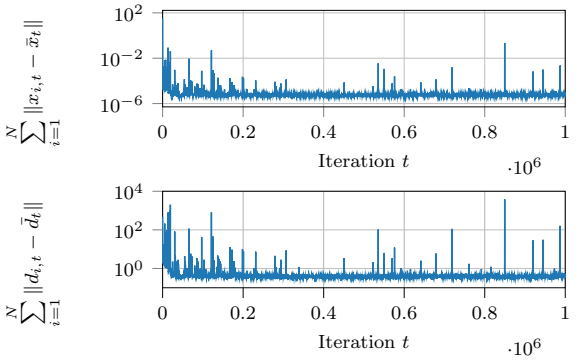

Fig. 3. Evolution of the consensus error (top) and the tracking error (bottom).

V. Conclusions

In this paper, we addressed the problem of solving in a distributed way an online optimization problems in which the local cost functions are composed by a known and an unknown part. We proposed an algorithm that concurrently tracks the solution of the problem and estimates the parameters of the unknown portion of the objective function. Finally, we showed that a bounded (possibly vanishing) average regret is achieved under suitable assumptions. A numerical example is provided to corroborate the theoretical results.

References

[1] F. Dressler, "Cyber physical social systems: Towards deeply integrated hybridized systems," in *International Conference on Computing, Networking and Communications (ICNC)*, 2018, pp. 420–424.

[2] P. Chatupromwong and A. Yokoyama, "Optimization of charging sequence of plug-in electric vehicles in smart grid considering user's satisfaction," in *IEEE International Conference on Power System Technology*, 2012, pp. 1–6.

[3] A. M. Ospina, A. Simonetto, and E. Dall'Anese, "Personalized demand response via shape-constrained online learning," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*, 2020.

[4] D. Quercia, R. Schifanella, and L. M. Aiello, "The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy Routes in the City," in *Conf. on Hypertext and Social Media*, 2014, pp. 116 – 125.

[5] M. Menner, L. Neuner, L. Lünenburger, and M. N. Zeilinger, "Using human ratings for feedback control: A supervised learning approach with application to rehabilitation," *IEEE Trans. on Robotics*, vol. 36, no. 3, pp. 789–801, 2020.

[6] X. Luo, Y. Zhang, and M. M. Zavlanos, "Socially-Aware Robot Planning via Bandit Human Feedback," in *ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, 2020, pp. 216–225.

[7] E. Dall'Anese, A. Simonetto, S. Becker, and L. Madden, "Optimization and learning with information streams: Time-varying algorithms and applications," *IEEE Signal Processing Magaz.*, vol. 37, pp. 71–83, 2020.

[8] A. Simonetto, E. Dall'Anese, J. Monteil, and A. Bernstein, "Personalized optimization with user's feedback," *Automatica*, vol. 131, p. 109767, 2021.

[9] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proc. of the IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.

[10] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, US: The MIT Press, 2006.

[11] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[12] R. Mazumder, A. Choudhury, G. Iyengar, and B. Sen, "A Computational Framework for Multivariate Convex Regression and Its Variants," *Journal of the American Statistical Ass.*, vol. 114, no. 525, pp. 318–331, 2019.

[13] A. Simonetto, "Smooth strongly convex regression," in *European Signal Processing Conference (EUSIPCO)*, 2021, pp. 2130–2134.

[14] A. L. Johnson and D. R. Jiang, "Shape constraints in economics and operations research," *Statistical Science*, pp. 527 – 546, 2018.

[15] W. Chen and R. Mazumder, "Multivariate Convex Regression at Scale," *Journal of the American Statistical Association*, 2020, arXiv:2005.11588.

[16] L. Ljung, *System Identification: Theory for the User*. Prentice Hall PTR, 1999.

[17] F. Hayashi, *Econometrics*. Princeton, NJ: Princeton Univ. Press, 2000.

[18] A. K. Sahu, S. Kar, J. M. F. Moura, and H. V. Poor, "Distributed Constrained Recursive Nonlinear Least-Squares Estimation: Algorithms and Asymptotics," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 426–441, 2016.

[19] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *IEEE International Symposium on Intelligent Control*, 2011, pp. 613–619.

[20] A. Flaxman, K. A.T., and H. McMahan, "Online Convex Optimization in the Bandit Setting: Gradient Descent without Gradient," in *Proc. of the ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 385 – 394.

[21] J. Duchi, M. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal Rates for Zero-Order Convex Optimization: The Power of Two Function Evaluations," *IEEE Trans. on Information Theory*, vol. 61, no. 5, pp. 2788 – 2806, 2015.

[22] F. Farina and G. Notarstefano, "Randomized block proximal methods for distributed stochastic big-data optimization," *IEEE Trans. on Automatic Control*, vol. 66, no. 9, pp. 4000–4014, 2021.

[23] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Mathematical Programming*, pp. 1–49, 2020.

[24] S. Rahili and W. Ren, "Distributed Convex Optimization for Continuous-Time Dynamics with Time-Varying Cost Functions," *IEEE Trans. on Automatic Control*, vol. 62, no. 4, 2017.

[25] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Trans. on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2017.

[26] M. Maros, "Distributed Optimization in Time-Varying Environments," Ph.D. dissertation, KTH - Stockholm, 2019.

[27] Q. Ling and A. Ribeiro, "Decentralized Dynamic Optimization Through the Alternating Direction Method of Multipliers," *IEEE Trans. on Signal Processing*, vol. 62, no. 5, pp. 1185–1197, 2014.

[28] A. S. Bedi, A. Koppel, and K. Rajawat, "Asynchronous Online Learning in Multi-Agent Systems With Proximity Constraints," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 5, no. 3, pp. 479–494, 2019.

[29] K. Yuan, W. Xu, and Q. Ling, "Can primal methods outperform primal-dual methods in decentralized dynamic optimization?" *IEEE Trans. on Signal Processing*, vol. 68, pp. 4466–4480, 2020.

[30] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," *IEEE Trans. on Automatic Control*, vol. 61, no. 4, pp. 994–1009, 2016.

[31] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[32] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[33] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2018.

[34] Y. Zhang, R. J. Ravier, M. M. Zavlanos, and V. Tarokh, "A distributed online convex optimization algorithm with improved dynamic regret," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 2449–2454.

[35] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *IEEE Conference on Decision and Control (CDC)*, 2016, pp. 7195–7201.

[36] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *European Journal of Control*, vol. 18, no. 6, pp. 539–557, 2012.

[37] I. Notarnicola, A. Simonetto, F. Farina, and G. Notarstefano, "Distributed personalized gradient tracking with convex parametric models," *arXiv preprint arXiv:2008.04363*, 2020.

[38] F. Farina, A. Camisa, A. Testa, I. Notarnicola, and G. Notarstefano, "DISROPT: a Python framework for distributed optimization," *21st IFAC World Congress*, 2020.

[39] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

APPENDIX

Let $\bar{x}_t \triangleq \frac{1}{N}\sum_{i=1}^N x_{i,t}$, $\bar{g}_t \triangleq \frac{1}{N}\sum_{i=1}^N \nabla \hat{f}_i(\bar{x}_t;t)$ and $\bar{d}_t \triangleq \frac{1}{N}\sum_{i=1}^N d_{i,t}$ be the averages of the local quantities in (9) for all $t \geq 0$. Through simple manipulations, we obtain

$$\bar{x}_t = \bar{x}_{t-1} - \alpha \bar{d}_{t-1}, \quad \bar{d}_t = \bar{d}_{t-1} + \frac{1}{N}\sum_{i=1}^N (g_{i,t} - g_{i,t-1}). \quad (13)$$

By exploiting the (column) stochasticity of the weights (cf. Assumption II.4), and the initialization $d_i^0 = \nabla \hat{f}_i(x_i^0;0)$ it can be shown that, for all $t \geq 0$,

$$\bar{d}_t = \frac{1}{N}\sum_{i=1}^N g_{i,t} = \frac{1}{N}\sum_{i=1}^N \nabla \hat{f}_i(x_{i,t},t). \quad (14)$$

Moreover, letting $\mathbf{x}_t \triangleq \mathrm{col}(x_{1,t},\dots,x_{N,t})$, $\mathbf{d}_t \triangleq \mathrm{col}(d_{1,t},\dots,d_{N,t})$ and $\mathbf{g}_t \triangleq \mathrm{col}(g_{1,t},\dots,g_{N,t})$, algorithm (9) can be restated as

$$\mathbf{x}_t = W\mathbf{x}_{t-1} - \alpha \mathbf{d}_{t-1}, \quad \mathbf{d}_t = W\mathbf{d}_{t-1} + (\mathbf{g}_t - \mathbf{g}_{t-1}). \quad (15)$$

where $W \triangleq \mathcal{W} \otimes I_n$ with $\otimes$ denoting the Kronecker product.

A. Intermediate Results

The analysis relies on the consensus error $\|\mathbf{x}_t - \mathbf{1}\bar{x}_t\|$, the tracking error $\|\mathbf{d}_t - \mathbf{1}\bar{d}_t\|$ and the optimality error $\|\bar{x}_t - x_\star(t)\|$, as presented in the next lemmas (proofs are provided in [37]).

Lemma A.1. *Let assumption II.4 hold. Then, for all $t \geq 0$,*

$$\|\mathbf{x}_t - \mathbf{1}\bar{x}_t\| \leq \sigma_W \|\mathbf{x}_{t-1} - \mathbf{1}\bar{x}_{t-1}\| + \alpha \|\mathbf{d}_{t-1} - \mathbf{1}\bar{d}_{t-1}\|$$

where σ_W be the spectral radius of $W - \frac{1}{N}\mathbf{1}\mathbf{1}$. □

Lemma A.2. *Let Assumptions II.1, II.2, II.4 hold. Then, for $t \geq \bar{t}$ and with probability $1-\delta$:*

$$\|\bar{x}_t - \hat{x}_\star(t)\| \leq \theta \|\bar{x}_{t-1} - \hat{x}_\star(t-1)\| + \alpha \tfrac{L}{\sqrt{N}} \|\mathbf{x}_{t-1} - \mathbf{1}\bar{x}_{t-1}\| + c_x$$

with $\theta = \max\{|1 - L\alpha/N|, |1 - m\alpha/N|\}$. □

Lemma A.3. *Let Assumptions II.1, II.2, II.4 hold. Then, for $t \geq \bar{t}$ and with probability $1-\delta$:*

$$\begin{aligned}\|\mathbf{d}_t - \mathbf{1}\bar{d}_t\| &\leq (\sigma_W + \alpha L)\|\mathbf{d}_{t-1} - \mathbf{1}\bar{d}_{t-1}\| + \\ &+ (L\|W - I\| + 2L + \alpha L^2\sqrt{N})\|\mathbf{x}_{t-1} - \mathbf{1}\bar{x}_{t-1}\| \\ &+ (2L\sqrt{N} + \alpha L^2\sqrt{N})\|\bar{x}_{t-1} - x_\star(t-1)\| + \sqrt{N}c_\nabla. \quad \square\end{aligned}$$

B. Proof of Theorem III.3

Let us define

$$\mathbf{v}_t := \begin{bmatrix} \|\bar{x}_t - \hat{x}_\star(t)\| \\ \|\mathbf{x}_t - \mathbf{1}\bar{x}_t\| \\ \|\mathbf{d}_t - \mathbf{1}\bar{d}_t\| \end{bmatrix}, \quad \mathbf{z} := \begin{bmatrix} c_\nabla \\ c_x \end{bmatrix}.$$

By combining Lemma A.1, A.2 and A.3, we have that

$$\mathbf{v}_t \leq A(\alpha)\mathbf{v}_{t-1} + B\mathbf{z} \quad (16)$$

for $t \geq \bar{t}$ and with probability $1-\delta$, where

$$A(\alpha) := \begin{bmatrix} \theta & \alpha\frac{L}{\sqrt{N}} & 0 \\ 0 & \sigma_W & \alpha \\ a_1 & a_2 & \sigma_W + \alpha L \end{bmatrix}, \quad B := \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ \sqrt{N} & 0 \end{bmatrix},$$

with $a_1 = \alpha L^2\sqrt{N} + 2L\sqrt{N}$ and $a_2 = L\|W - I\| + 2L + \alpha L^2\sqrt{N}$.

Now, since by assumption $\alpha \leq N/L$ and $m \leq L$, we have that $\theta = 1 - \alpha m/N$ and hence

$$\begin{aligned}A(\alpha) = &\begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma_W & 0 \\ 2L\sqrt{N} & L\|W - I\| + 2L & \sigma_W \end{bmatrix} \\ &+ \alpha \begin{bmatrix} -\frac{m}{N} & \frac{L}{\sqrt{N}} & 0 \\ 0 & 0 & 1 \\ L^2\sqrt{N} & L^2\sqrt{N} & L \end{bmatrix}.\end{aligned}$$

We use now [39, Theorem 6.3.12] for a small perturbation $\alpha > 0$. For $\alpha = 0$, the eigenvalues of $A(\alpha)$ are 1 and $\sigma_W < 1$. By continuity of the eigenvalues w.r.t. the matrix coefficients, for small enough α, the eigenvalues < 1 will remain < 1. For the single eigenvalue 1 with left eigenvector $\mathrm{col}(1,0,0)$ and right eigenvector $\mathrm{col}(1, 0, 2L\sqrt{N}/(1-\sigma_W))$, one can use [39, Theorem 6.3.12(i)], to say that the corresponding eigenvalue of $A(\alpha)$, say $\lambda(\alpha)$, will be $|\lambda(\alpha) - 1 + \alpha m/N| \leq \alpha\epsilon$ for any $\epsilon > 0$ and sufficiently small α. If then one selects e.g., $\epsilon = \frac{m}{2N}$, then $\lambda(\alpha) \in [1 - 3\alpha\frac{m}{2N}, 1 - \alpha\frac{m}{2N}]$, meaning that there exists a small enough α, for which all the eigenvalues of $A(\alpha)$ are all strictly less than one, and therefore the spectral radius of $A(\alpha)$, say ρ, becomes strictly less than one. Also, we notice that the input $\mathbf{z}$ is bounded. Since $\mathbf{v}_t, A, B, \mathbf{z}$ have nonnegative entries, we can expand (16) from $\bar{t}$ and get $\mathbf{v}_t \leq$

$A(\alpha)^{t-\bar{t}}\mathbf{v}_{\bar{t}} + \sum_{\tau=\bar{t}}^{t-1} A(\alpha)^{t-1-\tau} B\mathbf{z}$. Given Assumptions II.1-II.2, for any finite $\bar{t}$, $\|\mathbf{v}_{\bar{t}}\|$ is bounded. Therefore we can write

$$\|\mathbf{v}_t\| \leq \|A(\alpha)^{t-\bar{t}}\mathbf{v}_{\bar{t}}\| + \Big\| \sum_{\tau=\bar{t}}^{t-1} A(\alpha)^{t-1-\tau} B\mathbf{z} \Big\|$$
$$\leq \rho^{t-\bar{t}}\|\mathbf{v}_{\bar{t}}\| + \sum_{\tau=\bar{t}}^{t-1} \rho^{t-1-\tau}\sqrt{Nc_\nabla^2 + c_x^2}. \quad (17)$$

And, taking the limit superior:

$$\limsup_{t\to\infty} \|\mathbf{v}_t\| = \frac{1}{1-\rho}\sqrt{Nc_\nabla^2 + c_x^2} =: \bar{c}. \quad (18)$$

Eq. (17) shows that the first term decreases linearly with rate ρ equal to the spectral radius of $A(\alpha)$, while the second term is bounded. Eq. (18) completes the argument yielding the upper limit of the sequence. This finishes the first part of the proof.

The second part of the proof is based on similar arguments to those used in [33, Theorem 1]. In particular we have that all the entries of $\mathbf{v}_t$ converges to $\bar{c}$ linearly with rate $O(\rho^k)$. Moreover, by exploiting the Lipschitz continuity of the gradients of $\hat{f}(x;t) = \sum_i \hat{f}_i(x;t)$ one has $\hat{f}(\bar{x}_t;t) - \hat{f}(\hat{x}_\star(t);t) \leq \nabla\hat{f}(\hat{x}_\star(t);t)^\top(\bar{x}_t - \hat{x}_\star(t)) + \frac{L}{2}\|\bar{x}_t - \hat{x}_\star(t)\|^2$. Now, since $\nabla\hat{f}(\hat{x}_\star(t);t) = 0$ the above implies that $\hat{f}(\bar{x}_t;t) - \hat{f}_\star(t) \leq \frac{L}{2}\|\bar{x}_t - \hat{x}_\star(t)\|^2 \leq \frac{L}{2}\|\mathbf{v}_t\|^2$ and hence the $\limsup$ of $\hat{f}(\bar{x}_t;t) - \hat{f}_\star(t)$ converges linearly to $\frac{L\bar{c}^2}{2}$. It yields

$$\mathbf{Pr}\left(\limsup_{t\to\infty} \sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - \hat{f}_\star(t) = \frac{L(Nc_\nabla^2 + c_x^2)}{2(1-\rho)^2} \right) \geq 1-\delta.$$

This concludes the second part of the proof.

The third part concerns the convergence of the consensus metric C_T. By definition $C_T \leq \|\mathbf{v}_t\|^2$ so that the thesis follows.

The fourth part concerns the boundedness of $\|\bar{x}_t\|$, which is bounded by the discussion above as

$$\|\bar{x}_t\| \leq \|\bar{x}_t - \hat{x}_\star(t)\| + \|\hat{x}_\star(t)\| \leq \|\mathbf{v}_t\| + \|\hat{x}_\star(t)\|,$$

and since $\|\mathbf{v}_t\|$ is bounded for discussion above and $\|\hat{x}_\star(t)\|$ is finite by assumption, then $\|\bar{x}_t\|$ is bounded.

Finally, since the above limit results are valid for any $\delta \in (0,1]$, we have that they hold with high probability.

C. Asymptotical bounds for RLS

Next we present useful asymptotical bounds for RLS that are necessary for Theorem III.4. Their proofs are in [37].

Lemma A.4. *Assumption II.2(i) holds for our RLS scheme.*□

Lemma A.5. *For an estimator satisfying the assumptions of Lemma III.2, for any bounded vector $x \in \mathbb{R}^n$, the functional learning is bounded as $|\hat{U}_{i,t}(x) - U_i(x)| \leq O(1/\sqrt{t})$.* □

D. Proof of Theorem III.4

Recalling the definition of the cumulative dynamic regret in (2), we can write

$$R_T(\{\bar{x}_t\}_{t=1}^T) = \sum_{t=1}^T \left(\sum_{i=1}^N \Big(\hat{f}_i(\bar{x}_t;t) + U_i(\bar{x}_t) - \hat{U}_{i,t}(\bar{x}_t) \Big) - f_\star(t) \right)$$
$$\leq \sum_{t=1}^T \Big(\sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - f_\star(t) \Big) + \sum_{t=1}^T \sum_{i=1}^N \Big| U_i(\bar{x}_t) - \hat{U}_{i,t}(\bar{x}_t) \Big|$$

Now, fixing a $\delta \in (0,1]$ one determines a $\bar{t}$, and the first term on the right-hand side can be split as $\sum_{t=1}^T(\sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - f_\star(t)) = \sum_{t=1}^{\bar{t}-1}(\sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - f_\star(t)) + \sum_{t=\bar{t}}^T(\sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - f_\star(t))$, where in the first $\bar{t}$ iterations the functions $\hat{f}_i$, in general, could have been nonconvex, while they are convex after $\bar{t}$ with probability $1-\delta$. Notice now that by Assumptions II.1-II.2, both $V_i(x;t)$ and $\hat{U}_{i,t}(x)$ are bounded for all bounded x and all i and t. Moreover, by Theorem III.3, $\|\bar{x}_t\|$ is uniformly bounded. Thus, we can bound the quantity $\sum_{t=1}^{\bar{t}-1}\Big(\sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - f_\star(t)\Big)$ by $O(\bar{t})$. Then,

$$R_T(\{\bar{x}_t\}_{t=1}^T) \leq O(\bar{t}) + \sum_{t=\bar{t}}^T \Big(\sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - f_\star(t) \Big)$$
$$+ \sum_{t=1}^T \sum_{i=1}^N \Big| U_i(\bar{x}_t) - \hat{U}_{i,t}(\bar{x}_t) \Big|. \quad (19)$$

Now, we can use the fact that

$$f_\star(t) = \hat{f}_\star(t) + \underbrace{(\hat{f}(x_\star(t);t) - \hat{f}_\star(t))}_{(I)} + \underbrace{(f_\star(t) - \hat{f}(x_\star(t);t))}_{(II)}.$$

In addition, by strong convexity of $\hat{f}$ and optimality, it holds $(I) \geq \frac{m}{2}\|x_\star(t) - \hat{x}_\star(t)\|^2 \geq 0$, while $(II) \geq -\sum_{i=1}^N \Big| U_i(x_\star(t)) - \hat{U}_{i,t}(x_\star(t)) \Big|$. Putting these facts together in the expression of the dynamic regret (19), then,

$$R_T(\{\bar{x}_t\}_{t=1}^T) \leq O(\bar{t}) + O(c_U) + \sum_{t=\bar{t}}^T \Big(\sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - \hat{f}_\star(t) \Big). \quad (20)$$

Now, by using Theorem III.3, the second term can be upper bounded as

$$\sum_{t=\bar{t}}^T \Big(\sum_{i=1}^N \hat{f}_i(\bar{x}_t;t) - \hat{f}_\star(t) \Big)$$
$$\leq \sum_{t=\bar{t}}^T O(\rho^{t-\bar{t}}) + O\Big((T-\bar{t}) \frac{L(Nc_\nabla^2 + c_x^2)}{2(1-\rho)^2} \Big). \quad (21)$$

Hence, by combining (20) and (21) we have that, with probability $1-\delta$,

$$R_T(\{\bar{x}_t\}_{t=1}^T) \leq O(1) + O(c_U) + O\Big(T \frac{L(Nc_\nabla^2 + c_x^2)}{2(1-\rho)^2} \Big)$$

where we used the fact that, since $\bar{t}$ is finite, $O(\bar{t}) + \sum_{t=\bar{t}}^T O(\rho^{t-\bar{t}}) = O(1)$. Since the above is valid with probability $1-\delta$, for any $\delta \in (0,1]$, it is valid with high probability.

As for the consensus metric C_T, everything goes as in the proof of Theorem III.3, with the difference to be valid with high probability, which concludes the proof.