An online optimization approach for multi-agent tracking of dynamic parameters in the presence of adversarial noise

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.23919/ACC.2017.7963457">http://dx.doi.org/10.23919/ACC.2017.7963457</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers (IEEE)</td>
</tr>
<tr>
<td>Version</td>
<td>Author's final manuscript</td>
</tr>
<tr>
<td>Accessed</td>
<td>Sat Apr 13 05:42:41 EDT 2019</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/117776">http://hdl.handle.net/1721.1/117776</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Creative Commons Attribution-Noncommercial-Share Alike</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td><a href="http://creativecommons.org/licenses/by-nc-sa/4.0/">http://creativecommons.org/licenses/by-nc-sa/4.0/</a></td>
</tr>
</tbody>
</table>
An Online Optimization Approach for Multi-Agent Tracking of Dynamic Parameters in the Presence of Adversarial Noise

Shahin Shahrampour and Ali Jadbabaie

Abstract—This paper addresses tracking of a moving target in a multi-agent network. The target follows a linear dynamics corrupted by an adversarial noise, i.e., the noise is not generated from a statistical distribution. The location of the target at each time induces a global time-varying loss function, and the global loss is a sum of local losses, each of which is associated to one agent. Agents noisy observations could be nonlinear. We formulate this problem as a distributed online optimization where agents communicate with each other to track the minimizer of the global loss. We then propose a decentralized version of the Mirror Descent algorithm and provide the non-asymptotic analysis of the problem. Using the notion of dynamic regret, we measure the performance of our algorithm versus its offline counterpart in the centralized setting. We prove that the bound on dynamic regret scales inversely in the network spectral gap, and it represents the adversarial noise causing deviation with respect to the linear dynamics. Our result subsumes a number of results in the distributed optimization literature. Finally, in a numerical experiment, we verify that our algorithm can be simply implemented for multi-agent tracking with nonlinear observations.

I. INTRODUCTION

Distributed estimation, detection, and tracking is ubiquitous in engineering applications ranging from sensor and robotic networks to social networks, and it has received a lot of attention for many years [1]–[5]. In these scenarios, the task is to estimate the value of a parameter which may or may not be dynamic. A group of agents aim to accomplish this task as a team. Each individual agent only partially observes the parameter, but the global spread of observations in the network allows agents to estimate the parameter collaboratively. This would require agents to aggregate local information, and many methods use consensus protocols as a critical component [6]. It is well-known that when agents’ observations are linear with respect to the parameter, the tracking problem is equivalent to minimizing a global quadratic loss, written as a sum of local quadratic losses (see e.g. [7]). However, in general, the global loss can be more complicated, resulting in nonlinear observations.

In real-world applications, the parameter of interest is often time-varying. Therefore, regardless of the structure of the loss, the dynamic nature of the problem brings forward two issues: (i) The local losses are observed in an online or sequential fashion, i.e., the local losses are disclosed to agents only after they form their estimates at each round, and they are not aware of future loss functions. Therefore, the problem must be solved in an online setting. (ii) The online algorithm should mimic the performance of its offline counterpart in which the losses are known a priori. The gap between the two is often called regret. Tracking the minimizer of the global loss over time introduces the notion of dynamic regret [8]. This framework has been studied in centralized online optimization [8]–[12], where the hardness of the problem is captured via the variation in the minimizer sequence.

To address these issues in this paper, we adopt an online optimization approach to formulate distributed tracking. We consider tracking of a dynamic parameter or a moving target in a network of agents. The dynamics of the target is linear and known to agents, but the target deviates from this dynamics due to an unstructured or adversarial disturbance or noise. In other words, the noise is not necessarily generated from a statistical distribution, or it can be highly correlated to its past values over time. At each time instance, the target induces a global convex loss whose minimizer coincides with the target location. The global loss is a sum of local losses, where each local loss is associated to a specific agent. Agents exchange noisy local gradients according to a communication protocol to track the moving target.

Our problem setup is reminiscent of a distributed Kalman filter [13]. However, we differentiate the two as follows: (i) We do not assume that the target is driven by a Gaussian noise. Nor do we assume that this noise has a statistical distribution. Instead, we consider an adversarial-noise model with unknown structure. (ii) Agents observations are not necessarily linear; in fact, the observations are noisy local gradients that are non-linear when the loss is not quadratic. Furthermore, our focus is on the finite-time analysis rather than asymptotic results.

We propose a decentralized version of the Mirror Descent algorithm, developed by Nemirovski and Yudin [14]. Using the notion of Bregman divergence in lieu of Euclidean distance for projection, Mirror Descent has been shown to be a powerful tool in large-scale optimization. Our algorithm consists of three interleaved updates: (i) each agent follows the noisy local gradient while staying close to previous estimates in the local neighborhood; (ii) agents take into account the dynamics of the moving target; (iii) agents average their estimates in their local neighborhood in a consensus step.

We then use a dynamic notion of regret to measure the
The set \( \mathbb{R}^d \times d \times d \) is known, and \( v_t \in \mathbb{R}^d \) is an adversarial noise, i.e., the sequence \( \{ v_t \}_{t=1}^T \) is neither generated according to a statistical distribution, nor it is independent over time. Our goal is to track \( x^*_t \), and regardless of the observation model, a distribution-dependent mechanism, such as Kalman or particle filter, cannot solve the problem since the noise does not assume a statistical distribution.

In the centralized version of the tracking problem above, the observations of \( x^*_t \) are realized through a *time-varying, global* loss function. That is, the tracking problem above as an optimization, where \( x^*_t \) is the minimizer of the global loss at time \( t \). Let \( \mathcal{X} \) be a convex, compact set, and represent the global loss by \( f_t : \mathcal{X} \to \mathbb{R} \) at time \( t \). As the global loss varies over time, the goal is to track the minimizer of \( f_t(\cdot) \), which is \( x^*_t \). The *offline and centralized* version of our problem can be viewed as follows

\[
\begin{align*}
\text{minimize} & \quad \sum_{t=1}^T f_t(x_t) \\
\text{subject to} & \quad x_t \in \mathcal{X}, \ t \in [T].
\end{align*}
\]

We are interested to solve the problem above in an *online* and *decentralized* fashion. In particular, the global function at time \( t \) is a sum of \( n \) *local* functions as

\[
f_t(x) := \frac{1}{n} \sum_{i=1}^n f_{i,t}(x),
\]

where \( f_{i,t} : \mathcal{X} \to \mathbb{R} \) is a local convex function on \( \mathcal{X} \) for all \( i \in [n] \). We consider a network of \( n \) agents facing two challenges when solving problem (2): (i) agent \( j \in [n] \) receives information only about \( f_{j,t}(\cdot) \) and does not observe the global loss function \( f_t(\cdot) \), which is common to decentralized schemes; (ii) The functions are revealed to agents sequentially along the time horizon, i.e., at any time instance \( s \), agent \( j \) has observed \( f_{j,t}(\cdot) \) for \( t < s \), whereas the agent does not know \( f_{j,t}(\cdot) \) for \( s \leq t \leq T \), which is common to online settings.

The agents interact with each other, and their relationship is captured via an undirected graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \), where \( \mathcal{V} = [n] \) denotes the set of nodes, and \( \mathcal{E} \) is the set of edges. Each agent \( i \) assigns a positive weight \( [W]_{ij} \) for the information received from agent \( j \neq i \), and the set of neighbors of agent \( i \) is defined as \( \mathcal{N}_i := \{ j : [W]_{ij} > 0 \} \).

While the problem framework is reminiscent of a distributed Kalman [13], there are fundamental distinctions in our setup: (i) The adversarial noise \( v_t \) is neither Gaussian nor of known statistical distribution. It can be thought as a noise with unknown structure, which represents the deviation from the dynamic. (ii) Agents observations are not necessarily linear; in fact, the observations are local gradients of \( \{ f_{i,t}(\cdot) \}_{t=1}^T \) and are non-linear when the objective is not quadratic. The other implicit distinction in this work is our focus on *finite-time* analysis rather than asymptotic results.

From optimization perspective, our framework subsumes two important classes of decentralized optimization in the literature:

1) Existing methods often consider *time-invariant* objectives (see e.g. [15], [17], [24]). This is simply the special case where \( f_t(x) = f(x) \) and \( x_t = x \) in (2).

In online optimization, the focus is not on distribution of data. Instead, data is thought to be generated arbitrarily, and its effect is observed through the loss functions [23].
2) Online algorithms deal with *time-varying* functions, but often the network’s objective is to minimize the temporal average of \( \left\{ f_i(x) \right\}_{t=1}^T \) over a fixed variable \( x \) (see e.g. [18], [19]). This can be captured by our setup when \( x_t = x \) in (2).

However, in the tracking problem, *functions* and *comparator variables* evolve simultaneously, i.e., the variables \( \{x_t\}_{t=1}^T \) are not constrained to be fixed in (2). Recall that \( x^*_t := \text{argmin}_{x \in \mathcal{X}} f_t(x) \) is the minimizer of the global loss function at time \( t \). Then, the solution to problem (2) is simply \( \sum_{t=1}^T f_t(x^*_t) \). Denote by \( x_{i,t} \) the estimate of agent \( i \) for \( x^*_t \) at time \( t \). To exhibit the online nature of problem (2), we reformulate it using the notion of *dynamic regret* as follows:

\[
\text{Reg}^d_T = \frac{1}{n} \sum_{i=1}^n \left( \sum_{t=1}^T f_i(x_{i,t}) - \sum_{t=1}^T f_t(x^*_t) \right).
\]

(4)

Then, the objective is to minimize the dynamic regret above which measures the gap between the online algorithm and its offline version. Our performance bound shall exhibit the impact of system noise, i.e., we want to prove a regret bound in terms of

\[
\|v_t\| = \|x^*_{t+1} - Ax^*_t\|
\]

which represents the deviation of the moving target with respect to dynamics \( A \). Note that generalizing the results to the linear time-variant dynamics is straightforward, i.e., when \( A \) is replaced by \( A_t \) in (1).

**B. Technical Assumptions**

To solve the multi-agent online optimization (4), we propose to decentralize the Mirror Descent algorithm [14]. Mirror Descent has been shown to be a powerful method in large-scale optimization by using Bregman divergence in lieu of Euclidean distance in the projection step. Before defining Bregman divergence and elaborating the algorithm, we start by stating a couple of standard assumptions on loss functions and agents communication.

**Assumption 1:** For any \( i \in [n] \), the function \( f_{i,t}() \) is Lipschitz continuous on \( \mathcal{X} \) with a uniform constant \( L \). That is,

\[
|f_{i,t}(x) - f_{i,t}(y)| \leq L \|x - y\|,
\]

for any \( x, y \in \mathcal{X} \).

**Assumption 2:** The network is connected i.e., there exists a path from any agent \( i \in [n] \) to any agent \( j \in [n] \). Also, the matrix \( W \) is symmetric and doubly stochastic with positive diagonal. That is,

\[
\sum_{i=1}^n |W|_{ij} = \sum_{j=1}^n |W|_{ij} = 1.
\]

The connectivity constraint in Assumption 2 guarantees the information flow in the network.

We now outline the notion of Bregman divergence, which is critical in the development of Mirror Descent. Consider a compact, convex set \( \mathcal{X} \), and let \( \mathcal{R} : \mathcal{X} \rightarrow \mathbb{R} \) denote a 1-strongly convex function on \( \mathcal{X} \) with respect to a norm \( \| \cdot \| \). That is,

\[
\mathcal{R}(x) \geq \mathcal{R}(y) + \langle \nabla \mathcal{R}(y), x - y \rangle + \frac{1}{2} \|x - y\|^2.
\]

for any \( x, y \in \mathcal{X} \). Then, the Bregman divergence \( \mathcal{D}_{\mathcal{R}}(\cdot, \cdot) \) with respect to the function \( \mathcal{R}(\cdot) \) is defined as follows:

\[
\mathcal{D}_{\mathcal{R}}(x, y) := \mathcal{R}(x) - \mathcal{R}(y) - \langle x - y, \nabla \mathcal{R}(y) \rangle.
\]

The definition of the Bregman divergence and the strong convexity of \( \mathcal{R}(\cdot) \) imply that

\[
\mathcal{D}_{\mathcal{R}}(x, y) \geq \frac{1}{2} \|x - y\|^2,
\]

(6)

for any \( x, y \in \mathcal{X} \). Two famous examples of Bregman divergence are the Euclidean distance and the Kullback-Leibler (KL) divergence generated from \( \mathcal{R}(x) = \frac{1}{2} \|x\|^2_2 \) and \( \mathcal{R}(x) = \sum_{i=1}^d x(i) \log x(i) - x(i) \), respectively.

**Assumption 3:** Let \( x \) and \( \{y_i\}_{i=1}^n \) be vectors in \( \mathbb{R}^d \). We assume that the Bregman divergence satisfies the separate convexity in the following sense

\[
\mathcal{D}_{\mathcal{R}}(x, \sum_{i=1}^n \alpha(i) y_i) \leq \sum_{i=1}^n \alpha(i) \mathcal{D}_{\mathcal{R}}(x, y_i),
\]

where \( \alpha \in \Delta_n \) is on the \( n \)-dimensional simplex. The assumption is satisfied for commonly used cases of Bregman divergence. For instance, the Euclidean distance evidently respects the condition. The KL-divergence also satisfies the constraint, and we refer the reader to Theorem 6.4. in [25] for the proof.

**Assumption 4:** The Bregman divergence satisfies a Lipschitz condition of the form

\[
|\mathcal{D}_{\mathcal{R}}(x, z) - \mathcal{D}_{\mathcal{R}}(y, z)| \leq K \|x - y\|
\]

for all \( x, y, z \in \mathcal{X} \).

When the function \( \mathcal{R} \) is Lipschitz on \( \mathcal{X} \), the Lipschitz condition on the Bregman divergence is automatically satisfied. Again, for the Euclidean distance the assumption evidently holds. In the particular case of KL divergence, the condition can be achieved via mixing a uniform distribution to avoid the boundary (see e.g. [11] for more comments on the assumption).

**Assumption 5:** The dynamics \( A \) is assumed to be non-expansive. That is, the condition

\[
\mathcal{D}_{\mathcal{R}}(Ax, Ay) \leq \mathcal{D}_{\mathcal{R}}(x, y),
\]

holds for all \( x, y \in \mathcal{X} \), and \( \|A\| \leq 1 \).

The assumption postulates a natural constraint on the dynamics \( A \): it does not allow the effect of a poor estimation (at one step) to be amplified as the algorithm moves forward.
C. Decentralized Tracking via Online Mirror Descent

We now propose our algorithm to solve the problem formulated in terms of dynamic regret in \[7\]. In our setting, agents observations are gradients of the local losses. However, common in distributed state estimation and tracking, these observations are noisy. Hence, denoting the local gradient of agent \(i\) at time \(t\) by \(\nabla_{i,t} := \nabla f_i(x_{i,t})\), the agent only receives \(\nabla_{i,t}\) representing the stochastic gradient. The stochastic oracle that provides noisy gradients satisfies the following constraint \[7\]
\[
E[\|\nabla_{i,t} - \nabla_i\|^2] \leq L,
\]
where \(F_i\) is the \(\xi\)-field containing all information prior to the outset of round \(t + 1\). A commonly used model to generate stochastic gradients satisfying \[7\] is an additive, bounded, zero-mean noise. Agents then track the moving target using a decentralized variant of Mirror Descent as follows \[8\]
\[
\hat{x}_{i,t+1} = \arg\min_{x \in X} \left\{ \eta_t \langle x, \nabla_{i,t} \rangle + D_R(x, y_{i,t}) \right\},
\]
\[
x_{i,t} = A \hat{x}_{i,t}, \quad \text{and} \quad y_{i,t} = \sum_{j=1}^n [W]_{ij} x_{j,t},
\]
where \(\{\eta_t\}_{t=1}^T\) is the step-size sequence, and \(A \in \mathbb{R}^{d \times d}\) represents the estimate of agent \(i\) of the moving target \(x^*_t\) at time \(t\). The step-size sequence should be tuned for different cases, but it is generally non-increasing and positive.

The update \[8a\] allows an agent to follow the noisy local gradient while keeping the estimate close to those of the local neighborhood. This closeness occurs by minimizing the Bregman divergence. On the other hand, the first update in \[8a\] takes into account the dynamics of the moving target, and the second update in \[8b\] is the consensus term averaging the estimates in the local neighborhood.

III. Theoretical Results

In this section, we state our theoretical result on the non-asymptotic performance of the decentralized online mirror descent for tracking dynamic parameters. Theorem 1 proves a bound on the dynamic regret, which captures the deviation of the moving target from the dynamics \(A\) (tracking error), the decentralization cost (network error), and the impact of stochastic gradients (stochastic error). We show that this theorem recovers previous rates on decentralized optimization in dynamic setting when the network error is eliminated. The proof is given in Appendix (Section VI).

Theorem 1: Consider a moving target \(x^*_t \in \mathbb{R}^d\) with the dynamical model of \[1\]. Further consider the distributed, online tracking problem formulated in \[4\], where \(x_{i,t}\) denotes
\[3\]For simplicity, we use one constant \(L\) to bound gradients as well as the stochastic gradients.
\[4\]We set \(x_{i,t}\) to be the vector of all zeros to initialize the algorithm. In general, any initialization could work for the algorithm.

the local estimate of agent \(i\) of the moving target \(x^*_t\) at time \(t \in [T]\). Let the local estimates be generated by updates \[8a\]-\[8b\], where the stochastic gradients satisfy the condition \[7\]. Given Assumptions \[7\], the dynamic regret can be bounded as
\[
\text{Reg}^d_T \leq E_{\text{Track}} + E_{\text{Net}} + E_{\text{Stoch}},
\]
with probability at least \(1 - \delta\), where
\[
E_{\text{Track}} := \frac{2R^2}{\eta_{T+1}} + \sum_{t=1}^T \frac{K}{\eta_{t+1}} \|x^*_t - Ax^*_t\|^2 + L^2 \sum_{t=1}^T \frac{\eta_t}{2}
\]
\[
E_{\text{Net}} := 4L^2 \sqrt{n} \sum_{i=1}^n \sum_{\tau=0}^{T-1} \eta_t \sigma_2^{t-\tau-1}(W)
\]
\[
E_{\text{Stoch}} := 8L \sqrt{T} \log \delta,
\]
and \(R^2 := \sup_{x,y \in X} D_R(x,y)\).

In view of \[7\], the dynamical model of the target is described with the noise \(v_t\). The term \(E_{\text{Track}}\) shows the dependence of performance bound to noise by aggregating the errors \(\|x^*_{t+1} - Ax^*_t\|\) over time. Also, \(E_{\text{Net}}\) and \(E_{\text{Stoch}}\) are the errors related to network and stochastic gradients, respectively.

In Section II we discussed that our setup generalizes some of the previous results. It is now important to see that this generalization is valid in the sense that our result can recover those special cases:

- When the global loss \(f_t(x) = f(x)\) is time-invariant, the target \(\{x^*_t\}_{t=1}^T\) is fixed, i.e., the dynamics \(A = I_d\) and \(v_t = 0\) in \[7\]. In this case in Theorem 1 the term involving \(\|x^*_{t+1} - Ax^*_t\|\) in \(E_{\text{Track}}\) is equal to zero, and we can use the step-size sequence \(\eta = \sqrt{(1 - \sigma_2(W))/T}\) to recover the result of comparable algorithms, such as Theorem 4 in [17] on distributed dual averaging.
- The same argument holds when the global loss is time-invariant, but the target is fixed. This setup is studied, for instance, in [18] via distributed online dual averaging with exact gradients. Disregarding \(E_{\text{Stoch}}\) in our bound due to stochastic gradients, since \(\|x^*_{t+1} - Ax^*_t\| = 0\) again, we recover Corollary 3 in [18].
- When the graph is complete, \(\sigma_2(W) = 0\) and hence \(E_{\text{Net}} = 0\). We then recover the results of [9] on centralized online learning (for linear dynamics) with exact gradients once we remove \(E_{\text{Stoch}}\) due to stochastic gradients.

IV. Numerical Experiment: Tracking Maneuvering Targets

In Mirror Descent algorithm, one has freedom over the selection of the Bregman divergence. A particularly well-known type of Bregman is the Euclidean distance, commonly used in state estimation and tracking dynamic parameters. We focus on this scenario in this section to provide the numerical experiments for our method.

We consider a slowly maneuvering target in the 2D plane and assume that each position component of the target
evolves independently according to a near constant velocity model [26]. The state of the target at each time consists of four components: horizontal position, vertical position, horizontal velocity, and vertical velocity. We represent the state at time $t$ by $x^*_t \in \mathbb{R}^4$, and therefore, the state space model takes the form

$$x^*_{t+1} = Ax^*_t + v_t,$$

where $v_t \in \mathbb{R}^4$ is the system noise, and using $\otimes$ for Kronecker product, $A$ can be written as

$$A = I_2 \otimes \begin{bmatrix} 1 & \epsilon \\ 0 & 1 \end{bmatrix},$$

with $\epsilon$ being the sampling interval.\footnote{The sampling interval of $\epsilon$ (seconds) is equivalent to the sampling rate of $1/\epsilon$ (Hz).} The goal is to cooperatively track $x^*_t$ in a network of agents. This problem has been studied in the context of distributed Kalman filtering [13], [27], state estimation [28]–[30], and particle filtering [31]–[33]. However, in contrast to Kalman filtering, we do not assume that the system noise $v_t$ is Gaussian. Also, as opposed to particle filtering, we do not receive a large number of samples (particles) per iteration since our setup is online, i.e., agents only observe one sample per time. Furthermore, we do not assume a statistical distribution on $v_t$ in our analysis, which differentiates our framework from state estimation. We adopt a model-free approach where the noise can be adversarial (deterministic), stochastic with dependence over time, or of some complex structure. We generate the noise as follows. At each time $t$, the noise $v_t$ is generated from a zero-mean Gaussian distribution with covariance matrix $\Sigma$ as follows

$$\Sigma = \sigma^2 v I_2 \otimes \begin{bmatrix} \epsilon^2/3 & \epsilon^2/2 \\ \epsilon^2/2 & \epsilon \end{bmatrix},$$

for the sampling interval $\epsilon = 0.1$ seconds which amounts to frequency $10$ Hz. Then, we let the system noise be $v_t = \nu_t \|\nu_t\|_\infty$. Though $\nu_t$ is generated from Gaussian distribution, the mismatch noise $v_t$ is non-Gaussian and can have a complicated distribution. The constant $\sigma^2$ takes different values in each experiment, and we describe this choice later.

We consider a sensor network of $n = 25$ agents located on a $5 \times 5$ grid. Agents aim to track the moving target $x^*_t$ collaboratively. Agents observe a noisy version of the target through a local loss function, and these observations are nonlinear. In particular, let the quantity $z_{i,t}$ be a noisy version of one coordinate of $x^*_t$ as follows

$$z_{i,t} = \theta_k^T x^*_t + w_{i,t},$$

where $w_{i,t} \in \mathbb{R}$ denotes a random noise, and $\theta_k$ is the $k$-th unit vector in the standard basis of $\mathbb{R}^4$ for $k \in \{1, 2, 3, 4\}$. We partition the agents into four groups, and for each group we select one specific $k_i$ from the set $\{1, 2, 3, 4\}$. The random noise $w_{i,t}$ satisfies the standard assumption of being zero-mean and finite-variance. Again, to show that our results are not dependent on Gaussian noise, we generate $w_{i,t}$ independently from a uniform distribution on $[-1, 1]$.

Then, at time $t$ the local loss for agent $i$ takes the form

$$f_{i,t}(x) := \frac{1}{4} \mathbb{E} \left[ (z_{i,t} - \theta_k^T x)^4 \mid \mathcal{F}_{t-1}, x^*_t \right],$$

resulting in the global loss

$$f_t(x) := \frac{1}{4n} \sum_{i=1}^n \mathbb{E} \left[ (z_{i,t} - \theta_k^T x)^4 \mid \mathcal{F}_{t-1}, x^*_t \right],$$

where $\mathcal{F}_{t-1}$ is the $\sigma$-field containing all information in $\{w_{i,s} \}_{s=1}^{t-1}$. It is straightforward to see that $x^*_t$ is the minimizer of the global loss. Observation of agent $i$ at time $t$ is the stochastic gradient of the local loss

$$\nabla f_{i,t}(x) = (z_{i,t} - \theta_k^T x)^3 \theta_k.$$ We derive an explicit update to form an estimate $x_{i,t}$ of $x^*_t$. We use Euclidean distance as the Bregman divergence in updates \((8a), (8b)\) to get\footnote{We assume that the state of the target remains in a convex, compact set, and the updates can keep the estimate in the set without the projection step. This assumption can be satisfied in the finite-time domain.}

$$x_{i,t} = \sum_{j=1}^n \left[ W \right]_{ij} A x_{j,t-1} + \eta_k \theta_k \left( z_{i,t} - \theta_k^T x_{i,t-1} \right)^3,$$

tune and the step size to $\eta = \eta_t = 0.1$. The update is akin to consensus + innovation updates in the literature (see e.g. [2], [7]) though we recall that the observation is nonlinear, and the system noise $v_t$ is arbitrary. It is proved in [7] that in decentralized tracking, the dynamic regret can be presented in terms of the tracking error $x_{i,t} - x^*_t$ when the local losses are quadratic. More
specifically, the expected dynamic regret averages the tracking error over space and time (when normalized by $T$). While here we deal with polynomial loss of power four, the connection between tracking error and dynamic regret still holds true. Therefore, using the result of Theorem 1 we can expect that once the parameter does not deviate too much from the dynamics, i.e., when $\sum_{t=1}^{T} \|v_t\|$ is small, the bound on the dynamic regret as well as the collective tracking error is small.

![Fig. 2. The trajectory of $x_t^i$ over $T = 1000$ iterations is shown in red. We also depict the trajectory of the estimator $x_{i,t}$ (shown in blue) for $i \in \{1, 6, 12, 23\}$ and observe that it closely follows $x_t^i$ in every case.]

We show this intuitive idea by setting $\sigma^2$ to different values. Larger values for $\sigma^2$ are expected to cause more deviations from the dynamics $A$ and larger dynamic regret (worse performance). In Fig. 1, we plot the normalized dynamic regret for $\sigma^2 \in \{0.25, 0.5, 0.75, 1\}$. Note that for each value of $\sigma^2$, we run the experiment only once to investigate the high probability bound in Theorem 1. As expected, the performance improves once $\sigma^2$ tends to smaller values.

We next restrict our attention to the case that $\sigma^2 = 0.5$. For one run of this case, we provide a snapshot of the target trajectory (in red) in Fig. 2 and plot the estimator trajectory (in blue) for agents $i \in \{1, 6, 12, 23\}$. Fig. 2 suggests that agents’ estimators closely follow the trajectory of the moving target with high probability.

V. CONCLUSION

In this paper, we addressed tracking of a moving target in a network of agents. The target follows a linear dynamics which is common knowledge to agents, but it deviates from this dynamics due to an additive noise of an unknown structure. We formulated the problem as an online optimization of a global time-varying loss in a distributed fashion. The global loss at each time is a sum of a finite number of local losses, and each agent in the network holds a private copy of one local loss. Agents are unaware of the future loss functions as the local losses only become available to them sequentially. They exchange noisy local gradients with each other to track the value of the target.

Our proposed algorithm for this setup can be cast as a decentralized version of Mirror Descent. We however incorporated two more steps to include agents interactions and dynamics of the target. We used a notion of network dynamic regret to measure the performance of our algorithm versus its offline counterpart. We established that the regret bound scales inversely in the spectral gap of the network and captures the deviation of the target with respect to the dynamics. Our results generalized a number of results in online and offline distributed optimization. Also, numerical experiments verified the applicability of our algorithm to multi-agent tracking with nonlinear observations. Future directions include studying the algorithm in the case that several observations are available per round, i.e., when agents can receive multiple noisy gradients per time. The method can be useful in the sensor networks where each sensor can have multiple measurements from different sources.

REFERENCES


assume that the local functions are Lipschitz continuous (Assumption 2), the matrix \( W \) is doubly stochastic (Assumption 3), and the mapping \( A \) is non-expansive (Assumption 5). Then, the local estimates \( \{x_{i,t}\}_{t=1}^T \) generated by the updates (8a)-(8b) satisfy

\[
\|x_{i,t+1} - x_{t+1}\| \leq L\sqrt{\sum_{\tau=0}^{t} \eta_{\tau}\sigma^{2-\tau}}(W),
\]

for any \( i \in [n] \), where \( \tilde{x}_t := \frac{1}{n} \sum_{i=1}^{n} x_{i,t} \).

Lemma 3: Let \( X \) be a convex set in a Banach space \( B \), \( R : B \rightarrow \mathbb{R} \) denote a 1-strongly convex function on \( X \) with respect to a norm \( \| \cdot \| \), and \( D_R(\cdot, \cdot) \) represent the Bregman divergence with respect to \( R \), respectively. Furthermore, assume that the matrix \( W \) is doubly stochastic (Assumption 2), the Bregman divergence satisfies the Lipschitz condition and the separate convexity (Assumptions 3-4), and the mapping \( A \) is non-expansive (Assumption 5). Then, it holds that

\[
\frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} \left( \frac{1}{\eta_t} D_R(x_{i,t}^*, y_{i,t}) - \frac{1}{\eta_t} D_R(x_{i,t}^*, \hat{x}_{i,t+1}) \right) \leq 2R^2 \eta_{T+1} + \sum_{t=1}^{T} \frac{K}{\eta_{t+1}} \|x_{t+1}^* - Ax_t^*\|,
\]

where \( R^2 := \sup_{x,y \in X} D_R(x, y) \).

In what follows, we provide the proof of Theorem 1.

A. Proof of Theorem 7

Recall the definition of dynamic regret in (4). Using the Lipschitz continuity of \( f_i(\cdot) \) (Assumption 1) as well as the fact that the global loss is the sum of local losses (Eq. 4), we get

\[
\begin{align*}
& f_t(x_{i,t}) - f_t(x_{i,t}^*) = f_t(x_{i,t}) - f_t(\tilde{x}_t) + f_t(\tilde{x}_t) - f_t(x_{i,t}^*) \\
& \leq L \|x_{i,t} - \tilde{x}_t\| + f_t(\tilde{x}_t) - f_t(x_{i,t}^*) \\
& = L \|x_{i,t} - \tilde{x}_t\| + \frac{1}{n} \sum_{i=1}^{n} f_{i,t}(\tilde{x}_t) - \frac{1}{n} \sum_{i=1}^{n} f_{i,t}(x_{i,t}^*),
\end{align*}
\]

Using the Lipschitz continuity of \( f_{i,t}(\cdot) \) for \( i \in [n] \), we simplify above as follows

\[
\begin{align*}
& f_{i,t}(x_{i,t}) - f_{i,t}(x_{i,t}^*) \leq \frac{1}{n} \sum_{i=1}^{n} f_{i,t}(x_{i,t}) - \frac{1}{n} \sum_{i=1}^{n} f_{i,t}(x_{i,t}^*) \\
& + L \|x_{i,t} - \tilde{x}_t\| + \frac{L}{n} \sum_{i=1}^{n} \|x_{i,t} - \tilde{x}_t\|. \quad (9)
\end{align*}
\]

The second line can be controlled via Lemma 2 so we focus on the first term in the above bound. We have by convexity of \( f_{i,t}(\cdot) \) that

\[
\begin{align*}
f_{i,t}(x_{i,t}) - f_{i,t}(x_{i,t}^*) & \leq (\nabla_{i,t} x_{i,t}^* - x_{i,t}^*) \\
& = (\nabla_{i,t} x_{i,t} - x_{i,t}) + (\nabla_{i,t} - \nabla_{i,t} x_{i,t} - x_{i,t}^*) \\
& \leq (\nabla_{i,t} - \tilde{x}_{i,t+1} + x_{i,t}^*) + (\nabla_{i,t} - \tilde{x}_{i,t+1} + x_{i,t}^*) \\
& + (\nabla_{i,t} y_{i,t} - \tilde{x}_{i,t+1} + (\nabla_{i,t} - \nabla_{i,t} x_{i,t} - x_{i,t}^*), \quad (10)
\end{align*}
\]

VI. APPENDIX

We make use of two technical lemmas (Lemma 2 and 3) proved in the Appendix of [22]. We state their results here and use them in the proof of Theorem 1.

Lemma 2: Let \( X \) be a convex set in a Banach space \( B \), \( R : B \rightarrow \mathbb{R} \) denote a 1-strongly convex function on \( X \) with respect to a norm \( \| \cdot \| \), and \( D_R(\cdot, \cdot) \) represent the Bregman divergence with respect to \( R \), respectively. Furthermore,
We now need to bound each of the terms on the right hand side of (10). The stochastic gradients are bounded in view of (7). Therefore, using Hölder’s inequality for any primal-dual norm pair, we get
\[
\langle \nabla_{i,t}, y_{i,t} - \hat{x}_{i,t+1} \rangle \leq \|y_{i,t} - \hat{x}_{i,t+1}\| \|\nabla_{i,t}\|
\]
\[
\leq L \|y_{i,t} - \hat{x}_{i,t+1}\|
\]
\[
\leq \frac{1}{2\eta_t} \|y_{i,t} - \hat{x}_{i,t+1}\|^2 + \frac{\eta_t}{2} L^2,
\] (11)
where the last line is due to AM-GM inequality. We now recall update (8b) and use Assumptions 1 and 2 to derive
\[
\langle \nabla_{i,t}, \hat{x}_{i,t} - y_{i,t} \rangle = \langle \nabla_{i,t}, \hat{x}_{i,t} - \hat{x}_t + \hat{x}_t - y_{i,t} \rangle
\]
\[
= \langle \nabla_{i,t}, \hat{x}_{i,t} - \hat{x}_t \rangle + \sum_{j=1}^{n} [W]_{ij} \langle \nabla_{i,t}, \hat{x}_t - x_{j,t} \rangle
\]
\[
\leq L \|\hat{x}_{i,t} - \hat{x}_t\| + L \sum_{j=1}^{n} [W]_{ij} \|x_{j,t} - \hat{x}_t\|
\]
\[
\leq 2L^2 \sqrt{n} \sum_{\tau=0}^{t-1} \eta_{\tau}\sigma_2^{\tau-1}(W),
\] (12)
where in the last line we appealed to Lemma 2. Finally, the optimality of \(\hat{x}_{i,t+1}\) in (8a) implies (see e.g. Lemma 4.1 in [34]) that
\[
\langle \nabla_{i,t}, \hat{x}_{i,t+1} - x_t^* \rangle \leq \frac{1}{\eta_t} \mathcal{D}_R(x_t^*, y_{i,t}) - \frac{1}{\eta_t} \mathcal{D}_R(x_t^*, \hat{x}_{i,t+1})
\]
\[
- \frac{1}{\eta_t} \mathcal{D}_R(\hat{x}_{i,t+1}, y_{i,t})
\]
\[
\leq \frac{1}{\eta_t} \mathcal{D}_R(x_t^*, y_{i,t}) - \frac{1}{\eta_t} \mathcal{D}_R(x_t^*, \hat{x}_{i,t+1})
\]
\[
- \frac{1}{2\eta_t} \|\hat{x}_{i,t+1} - y_{i,t}\|^2,
\] (13)
since the Bregman divergence satisfies \(\mathcal{D}_R(x, y) \geq \frac{1}{2} \|x - y\|^2\) for any \(x, y \in \mathcal{X}\) in view of (6). Substituting (11), (12), and (13) into the bound (10), we derive
\[
f_{i,t}(x_{i,t}) - f_{i,t}(x_t^*) \leq \frac{\eta_t}{2} L^2 + 2L^2 \sqrt{n} \sum_{\tau=0}^{t-1} \eta_{\tau}\sigma_2^{\tau-1}(W)
\]
\[
+ \frac{1}{\eta_t} \mathcal{D}_R(x_t^*, y_{i,t}) - \frac{1}{\eta_t} \mathcal{D}_R(x_t^*, \hat{x}_{i,t+1})
\]
\[
+ \langle \nabla_{i,t} - \nabla_{i,t}, \hat{x}_{i,t} - x_t^* \rangle.
\] (14)
To bound the last term, we note that
\[
\mathbb{E} \left[ \langle \nabla_{i,t} - \nabla_{i,t}, \hat{x}_{i,t} - x_t^* \rangle \middle| \mathcal{F}_{t-1} \right]
\]
\[
= \mathbb{E} \left[ \langle \nabla_{i,t} - \nabla_{i,t}, \hat{x}_{i,t} - x_t^* \rangle \middle| \mathcal{F}_{t-1} \right], x_{i,t} - x_t^* = 0.
\]
Also, due to (6) we have \(\|x_{i,t} - x_t^*\|^2 \leq 2\mathcal{D}_R(x_{i,t}, x_t^*) \leq 2R^2\), which entails
\[
\langle \nabla_{i,t} - \nabla_{i,t}, \hat{x}_{i,t} - x_t^* \rangle \leq \|x_{i,t} - x_t^*\| \|\nabla_{i,t} - \nabla_{i,t}\| \leq 4LR.
\]
Therefore, summing \(\langle \nabla_{i,t} - \nabla_{i,t}, \hat{x}_{i,t} - x_t^* \rangle\) over \(t \in [T]\) forms a bounded difference martingale, and we can use Azuma’s inequality to get
\[
P \left( \sum_{i=1}^{n} \sum_{t=1}^{T} \langle \nabla_{i,t} - \nabla_{i,t}, \hat{x}_{i,t} - x_t^* \rangle \geq \epsilon \right) \leq e^{-\frac{\epsilon^2}{2\sum_{i=1}^{n} \sum_{t=1}^{T} \|\nabla_{i,t} - \nabla_{i,t}\|^2}}.
\]
Setting the probability above to \(\delta\) and solving for \(\epsilon\) implies
\[
\frac{1}{\eta_t} \sum_{i=1}^{n} \sum_{t=1}^{T} f_{i,t}(x_{i,t}) - f_{i,t}(x_t^*) \leq
\]
\[
L^2 \sum_{t=1}^{T} \frac{\eta_t}{2} + 2L^2 \sqrt{n} \sum_{t=1}^{T} \sum_{\tau=0}^{t-1} \eta_{\tau}\sigma_2^{\tau-1}(W)
\]
\[
+ \frac{1}{\eta_t} \sum_{i=1}^{n} \sum_{t=1}^{T} \left( \frac{1}{\eta_t} \mathcal{D}_R(x_t^*, y_{i,t}) - \frac{1}{\eta_t} \mathcal{D}_R(x_t^*, \hat{x}_{i,t+1}) \right)
\]
\[
+ 8LR \sqrt{-T \log \delta},
\]
with probability at least \(1 - \delta\). Applying Lemma 3 to above, we can simplify as
\[
\frac{1}{\eta_t} \sum_{i=1}^{n} \sum_{t=1}^{T} f_{i,t}(x_{i,t}) - f_{i,t}(x_t^*) \leq
\]
\[
L^2 \sum_{t=1}^{T} \frac{\eta_t}{2} + 2L^2 \sqrt{n} \sum_{t=1}^{T} \sum_{\tau=0}^{t-1} \eta_{\tau}\sigma_2^{\tau-1}(W)
\]
\[
+ \frac{2R^2}{\eta_{T+1}} + \frac{1}{\eta_{T+1}} \sum_{i=1}^{n} \sum_{t=1}^{T} \|x_{i,t} - Ax_t^*\|
\]
\[
+ 8L \sqrt{-T \log \delta}.
\]
We now return to sum (9) over \(t \in [T]\). We apply the bound above and the bound in Lemma 4 respectively to the first and second line in (9) to finish the proof.