REGRET BOUNDS OF A DISTRIBUTED SADDLE POINT ALGORITHM

Alec Koppel, Felicia Y. Jakubiec and Alejandro Ribeiro

Department of Electrical and Systems Engineering, University of Pennsylvania

ABSTRACT

An algorithm to learn optimal actions in distributed convex repeated games is developed. Learning is repeated because cost functions are revealed sequentially and distributed because they are revealed to agents of a network that can exchange information with neighboring nodes only. Learning is measured in terms of the global networked regret, which is the accumulated loss of causal prediction with respect to a centralized clairvoyant agent to which the information of all times and agents is revealed at the initial time. We use a variant of the Arrow-Hurwicz saddle point algorithm which penalizes local agent disagreement via Lagrange multipliers and leads to a distributed online algorithm. We show that decisions made with this saddle point algorithm lead to regret whose order is not larger than $O(\sqrt{T})$, where T is the total number of rounds of the game. Numerical behavior is illustrated for the particular case of dynamic sensor network estimation across different network sizes, connectivities, and topologies.

Index Terms— Distributed Statistical Learning, Online Convex Optimization, Sensor Network Estimation

1. INTRODUCTION

We consider learning problems in distributed repeated games. In this setup, for a given round of the game, each agent in a network makes a prediction, after which Nature reveals a loss function that measures its prediction quality. Moreover, individuals select actions with only access to local losses up to the previous time and information exchange with neighbors. The goal is for individuals to learn a strategy as good as one made by a player with access to complete information in advance at a central location.

Centralized repeated games may be formulated in the regret minimization framework [1, 2]. Here, a learner makes a sequence of plays to which Nature provides the answer in the form of a loss function. Regret is defined as the time accumulation of the loss difference between the online learner and a clairvoyant offline learner to which cost functions have been revealed beforehand. We interpret regret as a measure of the price for causal prediction. This paper makes use of online gradient descent whose regret that grows not faster than $O(\sqrt{T})$ and $O(\log T)$ for when losses convex and strongly convex, respectively [3]. Other methods to control regret growth are proximal maps [4], mirror descent [5, 6], and dual averaging [7], all of which may be understood as a special case of "follow the regularized leader" [1].

Strategies for solving distributed convex repeated games extend ideas from distributed optimization, which consider node-separable convex costs, and may be categorized into primal methods, dual methods, and primal-dual methods. In primal methods agents descend along their local gradients and average their signals with neighbors, [6, 8–10]. In dual methods agents reformulate distributed optimization as an agreement constrained optimization and use dual ascent, leveraging dual decomposition [11, 12]. Variations of dual methods include the alternating direction method of multipliers [13, 14] and newly developed second order methods [15]. Primal-dual methods combine primal descent with dual ascent [8, 16, 17]. Primal methods have been generalized to distributed online learning and have proved effective for particular cases where averaging is advantageous [18, 19]. A generalization of a primal dual method for distributed online learning comes in the form of the distributed saddle point algorithm (DSPA) [20, 21]. This paper provides theoretical regret bounds for DSPA (Section 4) as well as a numerical performance analysis (Section 5).

Thanks to NSF CCF-1017454, NSF CCF-0952867, & ONR N00014-12-1-0997.

2. REGRET IN DISTRIBUTED REPEATED GAMES

We formulate repeated games where at each round t predictions $\tilde{\mathbf{x}}_t \in X \subseteq \mathbb{R}^p$ are chosen by a player, after which instantaneous functions $f_t : \mathbb{R}^p \to \mathbb{R}^q$ are chosen by Nature that indicate quality of playing $\tilde{\mathbf{x}}_t$. In *offline* learning the functions f_t for times $t = 1, \dots T$ are known beforehand at time $t = 0$ and used to select a fixed strategy $\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}$ for all times. The total loss associated with the selection of $\tilde{\mathbf{x}}$ is $\sum_{t=1}^{T} f_t(\tilde{\mathbf{x}})$. In *online* learning the function f_t is revealed at time t and we are required to choose $\tilde{\mathbf{x}}_t$ without knowing f_t but rather the functions f_u that Nature played at earlier times $u < t$. The total loss associated with the variables $\tilde{\mathbf{x}}_t$ played for times $1 \leq t \leq T$ is the sum $\sum_{t=1}^{T} f_t(\tilde{\mathbf{x}}_t)$.

In this paper we are interested in cases in which functions f_t are written as a sum of components available at different agents in a network. Consider a symmetric and connected network $\mathcal{G} = (V, \mathcal{E})$ with N nodes forming the vertex set $V = \{1, \dots, N\}$ and $M = |\mathcal{E}|$ directed edges of the form $e = (j, k)$. Define the neighborhood of j as the set of nodes $n_j := \{k : (j, k) \in \mathcal{E}\}$ that share an edge with j. Each node in the network is associated with a sequence of cost functions $f_{i,t} : \mathbb{R}^p \to \mathbb{R}$ for all times $t \geq 1$. If a common variable $\tilde{\mathbf{x}}$ is played for all these functions the global network cost at time t is then given by

$$f_t(\tilde{\mathbf{x}}) = \sum_{i=1}^{N} f_{i,t}(\tilde{\mathbf{x}}). \tag{1}$$

and define the optimal offline strategy as $\tilde{\mathbf{x}}^* = \operatorname{argmin}_{\tilde{\mathbf{x}}} \sum_{t=1}^{T} f_t(\tilde{\mathbf{x}})$. The functions $f_{i,t}$ in (1), and as a consequence the functions f_t are assumed convex for all times t but are otherwise arbitrary.

Consider a coordinated game where all agents play a common variable $\tilde{\mathbf{x}}_t$ at time t. The accumulated regret associated with playing the coordinated sequence $\{\tilde{\mathbf{x}}_t\}_{t=1}^{T}$, as opposed to playing the optimal $\tilde{\mathbf{x}}^* = \operatorname{argmin}_{\tilde{\mathbf{x}}} \sum_{t=1}^{T} f_t(\tilde{\mathbf{x}})$ for all times t, can then be expressed as

$$\mathbf{Reg}_T^{\mathrm{C}} = \sum_{t=1}^{T} \sum_{i=1}^{N} f_{i,t}(\tilde{\mathbf{x}}_t) - \sum_{t=1}^{T} \sum_{i=1}^{N} f_{i,t}(\tilde{\mathbf{x}}^*). \tag{2}$$

In repeated games the goal is to design strategies that observe past functions f_u played by Nature at times $u < t$ to select and action $\tilde{\mathbf{x}}_t$ that makes the regret $\mathbf{Reg}_T^C$ in (2) small, i.e. $\mathbf{Reg}_T^C/T$ vanishes with growing T. The functions f_t are arbitrary and that while the offline strategy has the advantage of knowing all functions beforehand, online strategies have the advantage of being allowed to change their plays at each round.

An alternative formulation is to consider that agents play their own variables $\mathbf{x}_{i,t}$ to incur their own local cost $f_{i,t}(\mathbf{x}_{i,t})$. In this case we have the aggregate cost $\sum_{i=1}^{N} f_{i,t}(\mathbf{x}_{i,t})$ which leads to the definition of the uncoordinated regret by time T as

$$\mathbf{Reg}_T^{\mathrm{U}} = \sum_{t=1}^{T} \sum_{i=1}^{N} f_{i,t}(\mathbf{x}_{i,t}) - \sum_{t=1}^{T} \sum_{i=1}^{N} f_{i,t}(\tilde{\mathbf{x}}^*). \tag{3}$$

This formulation fails to incentivize agent collaboration: agents are effectively independent, and need only learn strategies that are good with respect to their local costs $\sum_{t=1}^{T} f_{i,t}(\mathbf{x}_{i,t})$(3). A simple local gradient

descent policy can achieve small regret with respect to the optimal *local* action $\mathbf{x}_i^* = \text{argmin}_{\mathbf{x}_i} \sum_{t=1}^T f_{i,t}(\mathbf{x}_i)$ [22].

A more appropriate formulation is to consider games where agents have an incentive to collaborate. Suppose then that each agent in the network plays his own variables $\mathbf{x}_{i,t}$ which may be different from the variables $\mathbf{x}_{j,t}$ played by other agents $j \neq i$ in the same round, yet still aims to learn a play that is optimal with respect to the global cost in (1). Thus, we formulate a problem in which the *local regret* of agent j is defined as

$$\mathbf{Reg}_T^j = \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\mathbf{x}_{j,t}) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\tilde{\mathbf{x}}^*). \tag{4}$$

The regret formulations in (2) and (4) correspond to problems in which agent j aspires to learn a play that is as good as the play that can be learned by a centralized agent that has access to the cost functions $f_{i,t}$ of all agents i. However, in the later the assumption is that only the local functions $f_{j,t}$ are known to agent j. We obtain a distributed repeated game which agent j successfully solves if $\mathbf{Reg}_T^j/T$ vanishes with increasing T. Average local regrets in (4) to define *global networked regret*

$$\mathbf{Reg}_T := \frac{1}{N}\sum_{j=1}^{N} \mathbf{Reg}_T^j = \frac{1}{N}\sum_{t=1}^{T}\sum_{i,j=1}^{N} f_{i,t}(\mathbf{x}_{j,t}) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\tilde{\mathbf{x}}^*), \tag{5}$$

where we used (4) and simplified terms to write the second equality. In this paper we develop a variation of the saddle point algorithm of Arrow and Hurwicz [16] to find a strategy whose local and global network regrets [cf. (4) and (5)] are of order not larger than $O(\sqrt{T})$. We also show that the proposed algorithm can be implemented by agents that have access to their local cost functions only and perform causal variable exchanges with neighbors. This saddle point algorithm is presented in Section 3.

3. ARROW-HURWICZ SADDLE POINT ALGORITHM

We turn to developing a saddle point algorithm to control the growth of the local and global network regrets [cf. (4) and (5)]. Since the regret functions $\mathbf{Reg}_T^j$ defined in (4) are the same for all agents j, plays $\mathbf{x}_{j,t}$ that are good for one agent are also good for another. Thus, a suitable strategy is to select actions $\mathbf{x}_{j,t}$ which are the same for every agent. Since the network $\mathcal{G}$ is assumed to be connected, this relationship can be attained by imposing the constraint $\mathbf{x}_{j,t} = \mathbf{x}_{k,t}$ for all pairs of neighboring nodes $(j,k) \in \mathcal{E}$. To write more compactly define the column vector $\mathbf{x}_t := [\mathbf{x}_{1,t}; \ldots; \mathbf{x}_{N,t}] \in \mathbb{R}^{Np}$ and the augmented graph edge incidence matrix $\mathbf{C} \in \mathbb{R}^{Mp \times Np}$. The matrix $\mathbf{C}$ is formed by $M \times N$ square blocks of dimension p. If the edge $e = (j,k)$ links node j to node k the block (e,j) is $[\mathbf{C}]_{ej} = \mathbf{I}_p$ and the block $[\mathbf{C}]_{ek} = -\mathbf{I}_p$, where $\mathbf{I}_p$ denotes the identity matrix of dimension p. All other blocks are identically null, i.e., $[\mathbf{C}]_{ek} = \mathbf{0}_p$ for all edges $e \neq (j,k)$. With this definitions the constraint $\mathbf{x}_{j,t} = \mathbf{x}_{k,t}$ for all pairs of neighboring nodes can be written as

$$\mathbf{C}\mathbf{x}_t = \mathbf{0}, \qquad \forall t = 1, \ldots, T. \tag{6}$$

The edge incidence matrix $\mathbf{C}$ has exactly p null singular values. We denote as $0 < \gamma$ the smallest nonzero singular value of $\mathbf{C}$ and as Γ the largest singular value of $\mathbf{C}$, both of which measure network connectedness.

Imposing the constraint in (6) for all times t requires global coordination – indeed, the formulation would be equivalent to the centralized regret problem in (2). Instead, we modify (1) to add a linear penalty term to incentivize the selection of coordinated actions. Introduce dual variables $\boldsymbol{\lambda}_{e,t} = \boldsymbol{\lambda}_{jk,t} \in \mathbb{R}^p$ associated with the constraint $\mathbf{x}_{j,t} - \mathbf{x}_{k,t} = \mathbf{0}$ and consider the addition of penalty terms of the form $\boldsymbol{\lambda}_{jk,t}^T(\mathbf{x}_{j,t} - \mathbf{x}_{k,t})$. For an edge that starts at node j, the multiplier $\boldsymbol{\lambda}_{jk,t}$ is assumed to be kept at node j. Denote the stacked vector $\boldsymbol{\lambda}_t := [\boldsymbol{\lambda}_{1,t}; \ldots; \boldsymbol{\lambda}_{M,t}] \in \mathbb{R}^{Mp}$ and define the online Lagrangian at time t as

$$\mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) \;=\; \sum_{i=1}^{N} f_{i,t}(\mathbf{x}_{i,t}) + \boldsymbol{\lambda}_t^T \mathbf{C}\mathbf{x}_t \;= f_t(\mathbf{x}) + \boldsymbol{\lambda}_t^T \mathbf{C}\mathbf{x}_t. \tag{7}$$

The definition in (7) corresponds to the Lagrangian associated with the minimization of the instantaneous function $\sum_{i=1}^N f_{i,t}(\mathbf{x}_{i,t})$ subject to the agreement constraint $\mathbf{C}\mathbf{x}_t = \mathbf{0}$. Using this online Lagragian we formulate the Arrow-Hurwicz saddle point method which exploits the fact that primal-dual optimal pairs are saddle points of the Lagrangian to work through successive primal/dual gradient descent/ascent steps, respectively.

Definition 1 *For the online Lagrangian in* (7) *the saddle point algorithm takes the form*

$$\mathbf{x}_{t+1} = \mathcal{P}_X[\mathbf{x}_t - \epsilon \nabla_{\mathbf{x}} \mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)], \tag{8}$$

$$\boldsymbol{\lambda}_{t+1} = \mathcal{P}_\Lambda[\boldsymbol{\lambda}_t + \epsilon \nabla_{\boldsymbol{\lambda}} \mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)], \tag{9}$$

where ϵ is a given stepsize, $\mathcal{P}_\Lambda(\boldsymbol{\lambda})$ denotes projection of dual variables on a given convex compact set Λ. The notation $\mathcal{P}_X(\mathbf{x})$ denotes projection onto the set of feasible primal variables so that we have $\mathbf{x}_j \in X$ for all the N components of the vector $\mathbf{x}_t := [\mathbf{x}_1; \ldots; \mathbf{x}_N]$.

We assume that the set of multipliers Λ can be written as a Cartesian product of sets Λ_{jk} so that the projection of $\boldsymbol{\lambda}$ into Λ is equivalent to the separate projection of the components $\boldsymbol{\lambda}_{jk}$ into the sets Λ_{jk}.

The pair of iterations in (8)-(9) can be implemented in a distributed manner such that the variables kept at node j, namely, $\mathbf{x}_{j,t}$ and $\boldsymbol{\lambda}_{jk,t}$, are updated using the values of other local variables and variables of neighboring nodes, namely, $\mathbf{x}_{k,t}$ and $\boldsymbol{\lambda}_{kj,t}$ for $k \in n_j$. See [20], Section III.

Proposition 1 *The updates in* (8)-(9) *may be separated along the components $\mathbf{x}_{j,t}$ associated with node j and Lagrange multiplier $\boldsymbol{\lambda}_{jk,t}$ associated with edge (j,k), yielding N and M respective parallel updates of the form*

$$\mathbf{x}_{j,t+1} = \mathcal{P}_X\Big[\mathbf{x}_{j,t} - \epsilon\Big(\nabla_{\mathbf{x}_j} f_{j,t}(\mathbf{x}_{j,t}) + \sum_{k \in n_j} (\boldsymbol{\lambda}_{jk,t} - \boldsymbol{\lambda}_{kj,t})\Big)\Big], \tag{10}$$

$$\boldsymbol{\lambda}_{jk,t+1} = \mathcal{P}_{\Lambda_{jk}}\Big[\boldsymbol{\lambda}_{jk,t} + \epsilon\,(\mathbf{x}_{j,t} - \mathbf{x}_{k,t})\Big], \tag{11}$$

where $\mathcal{P}_X(\mathbf{x}_{j,t})$ denotes projection of $\mathbf{x}_{j,t}$ into the feasible primal set X, and $\mathcal{P}_{\Lambda_{jk}}$ denotes projection of $\boldsymbol{\lambda}_{jk}$ into the dual set Λ_{jk}.

Node j can implement (10)-(11) by using local variables and receiving variables $\boldsymbol{\lambda}_{kj,t}$ and $\mathbf{x}_{k,t}$ maintained at neighboring nodes $k \in n_j$.

4. REGRET BOUNDS

We turn to establishing that the local and global network regrets in (4) and (5) associated with plays $\mathbf{x}_{j,t}$ generated by the saddle point algorithm in (8)-(9) grow not faster than $O(\sqrt{T})$. In order to obtain these results, some conditions are required of the primal and dual variables, cost functions, and network. We state these assumptions below.

(A1) The network $\mathcal{G}$ is connected. The smallest nonzero singular value of the incidence matrix $\mathbf{C}$ is γ, the largest singular value is Γ, and the network diameter is D.

(A2) The gradients of the loss functions for any $\mathbf{x}$ is bounded by a constant L, i.e.

$$\|\nabla f_t(\mathbf{x})\| \leq L. \tag{12}$$

(A3) The loss functions $f_{i,t}(\mathbf{x})$ are Lipschitz continuous with modulus $K_{i,t} \leq K$,

$$\|f_{i,t}(\mathbf{x}) - f_{i,t}(\mathbf{y})\| \;\leq\; K_{i,t}\|\mathbf{x} - \mathbf{y}\| \;\leq\; K\|\mathbf{x} - \mathbf{y}\|. \tag{13}$$

(A4) The set X of feasible plays is included in the 2-norm ball of radius $C_{\mathbf{x}}/N$.

$$X \;\subseteq\; \big\{\tilde{\mathbf{x}} \in \mathbb{R}^p : \|\tilde{\mathbf{x}}\| \leq C_{\mathbf{x}}/N\big\}. \tag{14}$$

(A5) The convex set Λ_{jk} onto which the dual variables $\boldsymbol{\lambda}_{jk,t}$ are projected is included in a 1-norm ball of radius $C_{\boldsymbol{\lambda}}$,

$$\Lambda_{jk} \;\subseteq\; \big\{\boldsymbol{\lambda} \in \mathbb{R}^p : \|\boldsymbol{\lambda}\|_1 \;\leq C_{\boldsymbol{\lambda}}\big\}, \tag{15}$$

for some constant $C_{\boldsymbol{\lambda}} \geq DNK + 1$.

Assumption (A1) is standard in distributed algorithms. Assumptions (A2)-(A3) are typical in the analysis of saddle point algorithms. The bounds on the sets X and Λ_{jk} in assumptions (A4)-(A5) are constructed so that the iterates $\mathbf{x}_{j,t}$ and $\boldsymbol{\lambda}_{jk,t}$ are bounded by the respective constants in (14) and (15). The constant $C_{\mathbf{x}}/N$ in Assumption 4 is chosen so that the 2-norm of the stacked primal iterates $\mathbf{x}_t := [\mathbf{x}_{1,t}; \dots; \mathbf{x}_{N,t}]$ are bounded as $\|\mathbf{x}_t\| \leq C_{\mathbf{x}}$.

The various bounds in Assumptions (A1) - (A5) permit bounding the norm of the gradients of the online Lagrangians in (7). For the gradient with respect to the primal variable $\mathbf{x}$, use of the triangle and Cauchy-Schwarz inequalities yields

$$\|\nabla_{\mathbf{x}} \mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\| = \|\nabla f_t(\mathbf{x}_t) + \mathbf{C}^T \boldsymbol{\lambda}_t\| \leq \|\nabla f_t(\mathbf{x}_t)\| + \|\mathbf{C}^T\| \|\boldsymbol{\lambda}_t\|. \quad (16)$$

Use now the bounds in (12) and (15) and the definition of Γ as the largest singular value of $\mathbf{C}$ to simplify (16) to

$$\|\nabla_{\mathbf{x}} \mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\| \leq L + \Gamma \sqrt{M} C_{\boldsymbol{\lambda}} := L_{\mathbf{x}}, \quad (17)$$

where we defined $L_{\mathbf{x}}$ for future reference. For the gradient with respect to the dual variable $\boldsymbol{\lambda}$, we can similarly write

$$\|\nabla_{\boldsymbol{\lambda}} \mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\| = \|\mathbf{C}\mathbf{x}_t\| \leq \|\mathbf{C}\| \|\mathbf{x}_t\| \leq \Gamma C_{\mathbf{x}} := L_{\boldsymbol{\lambda}}. \quad (18)$$

Our results concerning local and global networked regret are both derived from the following lemma that simultaneously bounds the uncoordinated regret in (3) and the weighted penalty disagreement $\sum_{t=1}^{T} \boldsymbol{\lambda}^T \mathbf{C} \mathbf{x}_t$.

Lemma 1 *Consider the sequence $\mathbf{x}_t := [\mathbf{x}_{1,t}; \dots; \mathbf{x}_{N,t}]$ generated by the saddle point algorithm in* (10)-(11)*. Let $\tilde{\mathbf{x}}^*$ be the optimal offline action in* (4)*, assume $\boldsymbol{\lambda}_1 = \mathbf{0}$ and further assume that assumptions 1 - 5 hold. If we select $\epsilon = 1/\sqrt{T}$ we have that for all $\boldsymbol{\lambda} \in \Lambda$ it holds*

$$\mathbf{Reg}_T^U + \sum_{t=1}^{T} \boldsymbol{\lambda}^T \mathbf{C} \mathbf{x}_t \leq \frac{\sqrt{T}}{2} \left(\|\mathbf{x}_1 - \tilde{\mathbf{x}}^*\|^2 + \|\boldsymbol{\lambda}\|^2 + L_{\mathbf{x}}^2 + L_{\boldsymbol{\lambda}}^2 \right). \quad (19)$$

From Lemma 1 we obtain a bound for the uncoordinated regret $\mathbf{Reg}_T^{\mathrm{U}}$ defined in (3). To do so note that $\boldsymbol{\lambda} = \mathbf{0} \in \Lambda$. Setting $\boldsymbol{\lambda} = \mathbf{0}$ in (19) yields

$$\mathbf{Reg}_T^{\mathrm{U}} \leq \frac{\sqrt{T}}{2} \left(\|\mathbf{x}_1 - \mathbf{x}\|^2 + L_{\mathbf{x}}^2 + L_{\boldsymbol{\lambda}}^2 \right). \quad (20)$$

This bound is of little use because, as we mentioned in Section 2, agents can reduce uncoordinated regret by just operating independently of each other. Observe, however, that the relationship in (19) also includes the weighted penalty disagreement $\sum_{t=1}^{T} \boldsymbol{\lambda}^T \mathbf{C} \mathbf{x}_t$. The presence of this term indicates that different users' actions can't be too far apart, from which we obtain a relationship between networked and uncoordinated regrets.

Theorem 1 *Let $\mathbf{x}_t := [\mathbf{x}_{1,t}; \dots; \mathbf{x}_{N,t}]$ denote the sequence generated by the saddle point algorithm in* (10)-(11) *and let $\tilde{\mathbf{x}}^*$ be the optimal offline action in* (4)*. If Assumptions 1-5 hold, with the initialization $\boldsymbol{\lambda}_1 = \mathbf{0}$ and step size $\epsilon = 1/\sqrt{T}$, the global network regret [cf.* (5)*] is bounded by*

$$\mathbf{Reg}_T \leq \frac{\sqrt{T}}{2} \left(\|\mathbf{x}_1 - \tilde{\mathbf{x}}^*\|^2 + M C_{\boldsymbol{\lambda}}^2 + L_{\mathbf{x}}^2 + L_{\boldsymbol{\lambda}}^2 \right) = O(\sqrt{T}). \quad (21)$$

Theorem 1 provides a guarantee that the saddle point iterates achieve a global networked regret that grows not faster than $O(\sqrt{T})$, which is the same as that of centralized problems with convex losses. The learning rate depends on primal initialization, network size and topology, as well as smoothness properties of the loss functions. The result established in Theorem 1 is a bound on the global networked regret which is the average the local regrets incurred by each agent. By relating the uncoordinated regret bound in (20) with the local regret defined in (4) we obtain a similar bound on the regret of each individual agent as we formally state next.

Theorem 2 *Let $\mathbf{x}_t := [\mathbf{x}_{1,t}; \dots; \mathbf{x}_{N,t}]$ be the sequence generated by the saddle point algorithm in* (10)-(11) *and let $\tilde{\mathbf{x}}^*$ be the global batch learner in* (4)*. If Assumptions 1-5 hold, with the initialization $\boldsymbol{\lambda}_1 = \mathbf{0}$ and step size $\epsilon = 1/\sqrt{T}$, the local regret of node j [cf.* (4)*] is bounded by*

$$\mathbf{Reg}_T^j \leq \frac{\sqrt{T}}{2} \left(\|\mathbf{x}_1 - \tilde{\mathbf{x}}^*\|^2 + M C_{\boldsymbol{\lambda}}^2 + L_{\mathbf{x}}^2 + L_{\boldsymbol{\lambda}}^2 \right) = O(\sqrt{T}). \quad (22)$$

Theorem 2 establishes that the local regret of each individual agent in the network grows at a rate not larger than $O(\sqrt{T})$, or that its time average vanishes as $O(1/\sqrt{T})$. It follows that individuals learn global information while only having access to local observations and the strategies of neighboring agents. The constants that bound the regret growth depend on the initial condition, network connectivity, and properties of the losses.

5. SIMULATION RESULTS

We formulate a dynamic estimation problem in a sensor network as a distributed repeated game using (5). Suppose each sensor in a network wants to estimate a signal $\tilde{\mathbf{x}} \in \mathbb{R}^p$ based upon observations $\mathbf{y}_{i,t} \in \mathbb{R}^q$. The signal $\tilde{\mathbf{x}}$ is related to the observations by $\mathbf{y}_{i,t} = \mathbf{H}_{i,t} \tilde{\mathbf{x}} + \mathbf{w}_{i,t}$, where the noise $\mathbf{w}_{i,t}$ is i.i.d. across sensor and time and follows a Gaussian distribution. The optimal estimator $\tilde{\mathbf{x}}^*$ given the observations $\mathbf{y}_{i,t}$ for all i and t is the least mean squared error estimator $\tilde{\mathbf{x}}^* = \operatorname{argmin}_x \sum_{t=1}^{T} \sum_{i=1}^{N} \|\mathbf{H}_{i,t} \tilde{\mathbf{x}} - \mathbf{y}_{i,t}\|^2$. If the signals $\mathbf{y}_{i,t}$ are known for all sensors i and times t the optimal estimator $\tilde{\mathbf{x}}^*$ can be easily computed. In this paper we are interested in cases where the signal $\mathbf{y}_{j,t-1}$ is revealed at time $t-1$ to sensor j which then proceeds to causally estimate the signal $\mathbf{x}_{j,t} \in \mathbb{R}^p$ based upon past observations $\{y_{j,u}\}_{u=1}^{t-1}$ and information received from neighboring nodes in previous time slots. Since signals are revealed sequentially to agents of a distributed network, this may be formulated as a distributed recursive least squares problem.

DSPA guarantees that the regret penalty $\mathbf{Reg}_T^j$ in (4) grows at a rate of $O(\sqrt{T})$ – see Sections 3 and 4. In (4), the local functions are $f_{i,t}(\mathbf{x}_{i,t}) = \|\mathbf{H}_{i,t} \mathbf{x}_{i,t} - \mathbf{y}_{i,t}\|^2$ and the primal update at agent j in (10) takes the form

$$\mathbf{x}_{j,t+1} = \mathcal{P}_{X_j} \Big[\mathbf{x}_{j,t} - \epsilon \Big(2\mathbf{H}_{j,t}^T \big(\mathbf{H}_{j,t} \mathbf{x}_{j,t} - \mathbf{y}_{j,t} \big) + \sum_{k \in n_j} \big(\boldsymbol{\lambda}_{jk,t} - \boldsymbol{\lambda}_{kj,t} \big) \Big) \Big]. \quad (23)$$

We consider the networked regrets in (4) and (5) as well as the relative error of the estimates $\mathbf{x}_{j,t}$ with respect to the optimal batch estimator $\tilde{\mathbf{x}}^*$ and the relative agreement between estimates $\mathbf{x}_{j,t}$ and $\mathbf{x}_{k,t}$ of different agents. Specifically, the relative error associated with the estimate $\mathbf{x}_{j,t}$ of agent j at time t is defined as

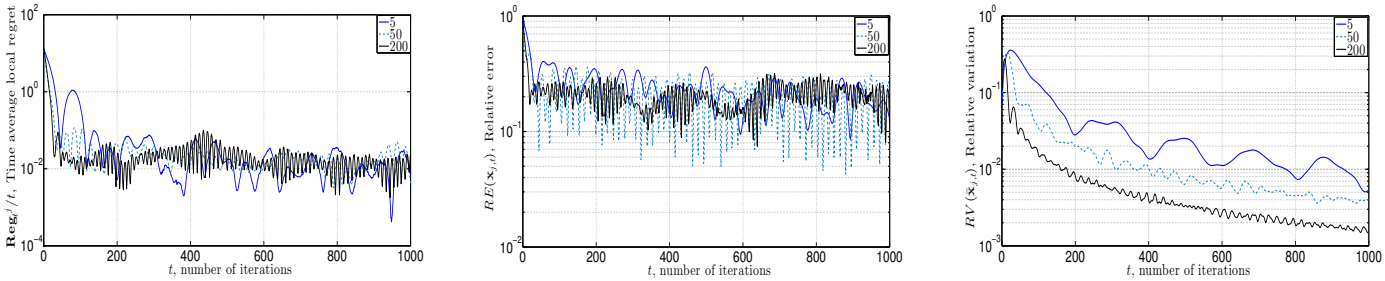
$$\mathrm{RE}(\mathbf{x}_{j,t}) := \|\mathbf{x}_{j,t} - \tilde{\mathbf{x}}^*\| / \|\tilde{\mathbf{x}}^*\|. \quad (24)$$

The agreement between predictions of different agents is defined in terms of the variable time averages $\bar{\mathbf{x}}_{j,t} := (1/t) \sum_{u=1}^{t} \mathbf{x}_{j,u}$. For the average estimate $\bar{\mathbf{x}}_{j,t}$ of agent j at time t we define the average relative variation as

$$\mathrm{RV}(\bar{\mathbf{x}}_{j,t}) := (1/N) \sum_{k=1}^{N} \|\bar{\mathbf{x}}_{j,t} - \bar{\mathbf{x}}_{k,t}\| / \|\tilde{\mathbf{x}}^*\|. \quad (25)$$

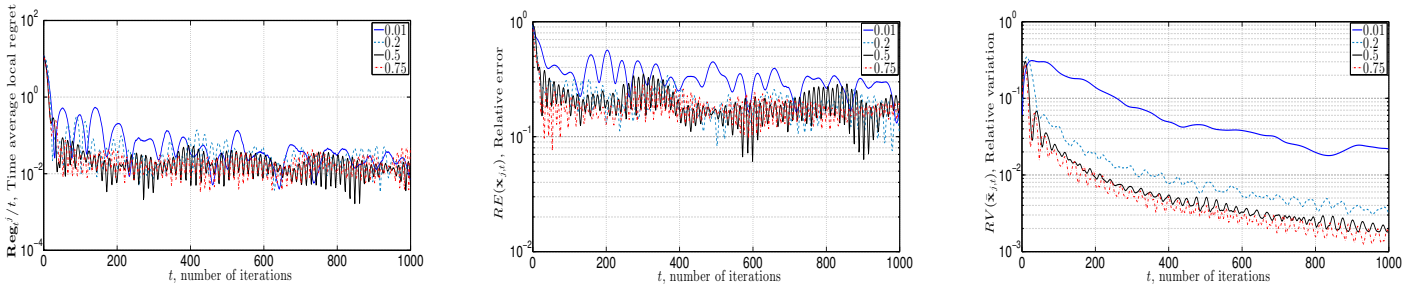
The average relative variation $\mathrm{RV}(\bar{\mathbf{x}}_{j,t})$ denotes the average Euclidean error between $\bar{\mathbf{x}}_{j,t}$ and all others, relative to the magnitude of the offline strategy $\tilde{\mathbf{x}}^*$. The reason to focus on time averages $\bar{\mathbf{x}}_{j,t}$ instead of the estimates $\mathbf{x}_{j,t}$ is that the latter oscillate around the batch estimate $\tilde{\mathbf{x}}^*$ and agreement between estimates of different agents is difficult to visualize.

We study the effect of network size, connectivity, and topology on the learning rates established in Section 4. Consider the scalar observation case $\mathbf{y}_{it} = \mathbf{H}_{i,t} \tilde{\mathbf{x}} + \mathbf{w}_{i,t}$ ($q = 1$) with signal $\tilde{\mathbf{x}}$ dimension $p = 10$. The matrices $\mathbf{H}_{i,t} = \mathbf{H}_i \in \mathbb{R}^{1 \times p}$ are constant across time but vary across agents, whose components are chosen equiprobably from $\{1/p, 2/p, \dots, 1\}$. Noise terms are sampled as $\mathbf{w}_{i,t} \sim \mathcal{N}(0, 0.1)$ with true signal $\tilde{\mathbf{x}} = \mathbf{1}$. We run (23) - (11) for $T = 10^3$ total iterations with step size $\epsilon = 1/\sqrt{T} = 0.03$ with initializations $\mathbf{x}_{j,1} = \mathbf{0}$ for all j and



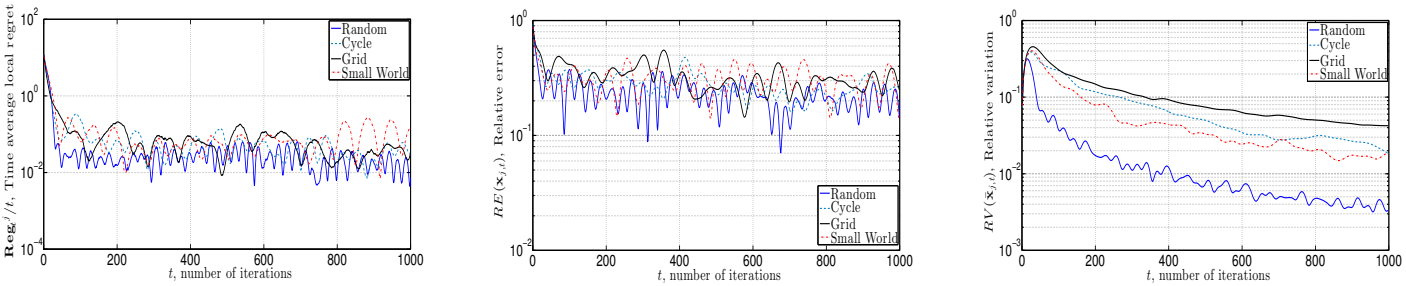
(a) Average regret $\mathbf{Reg}_t^j/t$ versus iteration t (b) Relative error $\text{RE}(\mathbf{x}_{j,t})$ versus iteration t (c) Relative variation $\text{RV}(\bar{\mathbf{x}}_{j,t})$ versus iteration t

Fig. 1: Learning achieved by an arbitrary agent in networks of size $N = 5$, $N = 50$, and $N = 200$ with nodes randomly connected with prob. $\rho = 0.2$. 1a-1b show $\mathbf{Reg}_t^j/t$ and $\text{RE}(\mathbf{x}_{j,t})$ versus iteration t, both of which decline and are less stable in smaller networks. Figure 1c shows that network disagreement in terms of $RV(\bar{\mathbf{x}}_{j,t})$ becomes more stable and declines faster with increasing N, as information contained per individual declines.



(a) Average regret $\mathbf{Reg}_t^j/t$ versus iteration t (b) Relative error $\text{RE}(\mathbf{x}_{j,t})$ versus iteration t (c) Relative variation $\text{RV}(\mathbf{x}_{j,t})$ versus iteration t

Fig. 2: Algorithm performance on a random $N = 50$ node network with connection probability $\rho \in \{0.01, 0.2, 0.5, 0.75\}$. Figure 2a-2b show $\mathbf{Reg}_T^j/T$ and $\text{RE}(\mathbf{x}_{j,t})$ as compared with iteration t for an arbitrary agent j. Both $\mathbf{Reg}_T^j/T$ and $\text{RE}(\mathbf{x}_{j,t})$ are more oscillatory in less connected networks. Figure 2c shows $RV(\bar{\mathbf{x}}_{j,t})$ versus iteration t. Primal variable consensus is more difficult to achieve in networks with fewer communication links.



(a) Average regret $\mathbf{Reg}_t^j/t$ versus iteration t (b) Relative error $\text{RE}(\mathbf{x}_{j,t})$ versus iteration t (c) Relative variation $\text{RV}(\bar{\mathbf{x}}_{j,t})$ versus iteration t

Fig. 3: Algorithm performance in $N = 50$ node cycle, grid, random and small world networks, where edges are generated randomly between agents with probability $\rho = 0.2$ in the later two. Figure 3a-3b show $\mathbf{Reg}_T^j/T$ and $\text{RE}(\mathbf{x}_{j,t})$, respectively, over iteration t and demonstrate that learning slows and oscillations increase with increasing network diameter. Figure 3c shows that in terms of $\text{RV}(\bar{\mathbf{x}}_{j,t})$, agents reach consensus slower with larger diameter.

$\boldsymbol{\lambda}_{jk,1} = \mathbf{0}$ for all j and k. The subsequent analysis uses an arbitrarily chosen agent in the network.

To investigate the role of the network size N on the learning rate we run (23) - (11) for problems with $N = 5$, $N = 50$, and $N = 200$ agents, where nodes are connected with probability with $\rho = 0.2$. The results of this experiment are given in Figure 1. Figure 1a shows $\mathbf{Reg}_t^j/t$ over iteration t, which declines at comparable rates across varying N. This rate similarity is also reflected in the trajectory of $\text{RE}(\mathbf{x}_{j,t})$ over time t, as shown in Figure 1b. In Figure 1b, we see that learning is comparable across different networks sizes but less stable with smaller N. This stability difference reflects the fact that as N increases, the relative information per agent decreases. Figure 1c shows that the network reaches consensus faster with larger N, as measured with $\text{RV}(\bar{\mathbf{x}}_{j,t})$, i.e. $\text{RV}(\bar{\mathbf{x}}_{j,t}) \leq 10^{-2}$, the algorithm requires $t = 719$, $t = 317$, and $t = 179$ iterations for $N = 5$, $N = 50$, and $N = 200$ node networks, respectively, suggesting the agreement constraint plays a larger role in larger networks.

To study the impact of network connectivity on learning, we fix the network size to $N = 50$ and run (23) - (11) on random networks with node connection probability $\rho \in \{0.01, 0.2, 0.5, 0.75\}$, the results of which are given in Figure 2. Figure 2a shows $\mathbf{Reg}_t^j/t$ versus iteration t. Here connectivity differences leads to comparable learning rates yet the numerical stability varies substantially. The sparsely connected networks oscillate more, as may be observed in the relative error plot in Figure 2b, which follows from the slower diffusion of information. Figure 2c shows the evolution of $\text{RV}(\bar{\mathbf{x}}_{j,t})$ over time, which demonstrates that trimal variable consensus is challenging to achieve in less connected networks.

To study the role of network topology in learning, we fix the network size to $N = 50$ and run (23) - (11) on random graphs, small world graphs, cycles, and grids, the results of which are shown in Figure 3. In the first two, the probability that node pairs are connected is fixed at $\rho = 0.2$. The latter two are deterministically generated; see [23, 24]. In Figure 3a, we plot $\mathbf{Reg}_t^j/t$ compared with iteration t. To surpass $\mathbf{Reg}_t^j/t \leq 10^{-2}$, we require $t = 293$, $t = 221$ iterations for random and small world networks, respectively, whereas for grids and cycles it requires $t = 483$, $t = 865$ iterations. As seen in the plot of $\text{RE}(\mathbf{x}_{j,t})$ over round t in Fig 3b, to attain $\text{RE}(\mathbf{x}_{j,t}) \leq 0.2$ we require $t = 81$, $t = 176$, $t = 556$, and $t = 578$ iterations for random, small world, grid, and cycle networks, respectively. Fig. 3c plots $\text{RV}(\bar{\mathbf{x}}_{j,t})$ over time t. To obtain $\text{RV}(\bar{\mathbf{x}}_{j,t}) \leq 5 \times 10^{-2}$, $t = 49$, $t = 301$, $t = 809$, and $t = 525$ iterations are required for random, small world, grid, and cycle networks, respectively. Structured deterministic networks are a more difficult setting, and the randomness in random and small world networks allows more effective information flow.

6. REFERENCES

[1] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1561/2200000018

[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[3] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. on Machine Learning*, vol. 20, no. 2, Washington DC, USA, Aug. 21-24 2003, pp. 928–936.

[4] C. B. Do, Q. V. Le, and C. Foo, "Proximal regularization for online and batch learning." in *ICML*, ser. ACM International Conference Proceeding Series, A. Danyluk, L. Bottou, and M. Littman, Eds., vol. 382. ACM, 2009, p. 33. [Online]. Available: http://dblp.uni-trier.de/db/conf/icml/icml2009.html#DoLF09

[5] S. Shalev-shwartz and Y. Singer, "Logarithmic regret algorithms for strongly convex repeated games," in *The Hebrew University*, 2007.

[6] S. Ram, A. Nedic, and V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[7] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, vol. 28, no. 1. Atlanta, GA: JMLR Workshop and Conference Proceedings, 2013, pp. 392–400. [Online]. Available: http://jmlr.csail.mit.edu/proceedings/papers/v28/suzuki13.pdf

[8] A. Nedic and A. Ozdaglar, "Subgradient methods for saddle-point problems," *J Optimiz. Theory App.*, vol. 142, no. 1, pp. 205–228, Aug. 2009.

[9] D. Jakovetic, J. M. F. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *CoRR*, vol. abs/1112.2972, Apr. 2011.

[10] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *ArXiv e-prints 1310.7063*, Oct. 2013.

[11] M. Rabbat, R. Nowak, and J. Bucklew, "Generalized consensus computation in networked systems with erasure links," in *IEEE 6th Workshop Signal Process. Adv. in Wireless Commun Process.*, Jun. 5-8 2005, pp. 1088–1092.

[12] F. Jakubiec and A. Ribeiro, "D-map: Distributed maximum a posteriori probability estimation of dynamic systems," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 450–466, Feb. 2013.

[13] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc wsns with noisy links - part i: distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[14] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "Dlm: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process.*, Aug. 2014.

[15] M. Zargham, A. Ribeiro, A. Jadbabaie, and A. Ozdaglar, "Accelerated dual descent for network optimization," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 905 – 920, Apr. 2014.

[16] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in linear and non-linear programming*, ser. With contributions by H. B. Chenery, S. M. Johnson, S. Karlin, T. Marschak, R. M. Solow. Stanford Mathematical Studies in the Social Sciences, vol. II. Stanford University Press, Stanford, 1958.

[17] Y. Nesterov, "Primal-dual subgradient methods for convex problems," Tech. Rep., 2005.

[18] K. I. Tsianos and M. G. Rabbat, "Distributed strongly convex optimization," *CoRR*, vol. abs/1207.3031, 2012.

[19] F. Yan, S. V. N. Vishwanathan, and Y. Qi, "Cooperative autonomous online learning," *CoRR*, vol. abs/1006.4039, 2010.

[20] A. Koppel, F. Jakubeic, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, Sep. 2014.

[21] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization." in *Proc. Int. Conf. Acoust. Speech Signal Process.*, May 4-9 2014, pp. 8292–8296.

[22] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization," *LIDS Report*, vol. 2800, 2009.

[23] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[24] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, Sept. 2000.