2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)

A SADDLE POINT ALGORITHM FOR NETWORKED ONLINE CONVEX OPTIMIZATION

Alec Koppel, Felicia Y. Jakubiec and Alejandro Ribeiro

Department of Electrical and Systems Engineering, University of Pennsylvania

ABSTRACT

This paper considers an online convex optimization problem in a distributed setting, where a connected network collectively solves a learning problem while only exchanging information between neighboring nodes. We formulate two expressions to describe distributed regret and present a variant of the Arrow-Hurwicz saddle point algorithm to solve the distributed regret minimization problem. Using Lagrange multipliers to penalize the discrepancy between them, only neighboring nodes exchange decision values and Lagrange multipliers. We show that decisions made with this saddle point algorithm lead to vanishing regret of the order of $O(1/\sqrt{T})$ where T is the final iteration time, and further depends on the smoothness of the cost functions and the size and connectivity of the network. Using a recursive least squares example, we find that the numerical results corroborate our theoretical findings.

1. INTRODUCTION

We consider the problem of distributed online learning within a network, in particular online convex optimization in which loss functions are convex. The goal is for each node in the network to learn global information while making autonomous decisions but only having access to partial information of the network, i.e. there are only information exchanges between neighbors. To meet this goal, we present a solution using saddle point algorithm relying upon a primal-dual subgradient descent-ascent concept.

In centralized online learning, gradient-based methods are well understood. With other similar methods such as proximal methods [1, 2], dual averaging [3] and the mirror descent algorithm [4, 5], online gradient descent can be understood as a special case of the FTRL (follow the regularized leader) algorithm, for an overview connecting these algorithms see [6]. As a result, all these algorithm yield $O(1/\sqrt{T})$ regret for convex functions. For some algorithms, including online gradient descent, additional smoothness assumptions such as strong convexity achieve $O(\log T/T)$ regret.

Distributed online learning for convex problems have been primarily approached based on variations of online gradient descent. Previous work use the consensus protocol in addition to a gradient descent step, which adds weighted values of neighboring nodes to generate a decision accounting for the information from neighbors and over time leads all nodes to reach a network-wide agreement on the learning result. [7, 8] As an alternative to such consensus-based methods, we propose a saddle point algorithm using the gradients of an online Lagrangian which allows deviations from the agreement and uses prices to penalize such deviations, thus avoiding the fallacies encountered in consensus algorithms. Equivalent to the Arrow-Hurwicz algorithm originating from [9], we are specifically interested in applying the saddle point method to Lagrangian relaxation problems, see [10].

Section 2 starts by introducing the concept of regret minimization for networked online optimization, for which we present two distributed regret formulations and compare them to the centralized online optimization problem. Section 3 develops the saddle point algorithm solving the networked online optimization problem by introducing an online Lagrangian whose gradient will be used to update primal and dual variables. In order to show that the algorithm from Section 3 minimizes regret, we show vanishing regret bounds in the order $O(1/\sqrt{T})$ in Section 4. Section 5 formulates the algorithm for a distributed recursive least squares problem and provides a numerical example which matches our theoretical performance results.

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

2. REGRET MINIMIZATION FOR DISTRIBUTED LEARNING

Consider the general problem of online learning, in which the learner is faced with a set of "answers" to a given question at each time t and is required to pick one, which we denote by $\mathbf{x}_t$, a vector of size J. Afterwards, he receives information about the gain or loss of his choice. The game-theoretic interpretation of this setting is a two-player game in which the learner plays against Nature which chooses $\mathbf{f}_t$ at each time step t such that the loss incurred by the learner is described by a loss function $l_t(\mathbf{x}_t, \mathbf{f}_t)$. If the learner follows a specific rule or online algorithm according to which he makes his choices, then the cumulative loss over time $\sum_{t=1}^{T} l_t(\mathbf{x}_t, \mathbf{f}_t)$ assesses the quality of the algorithm.

Regret is a quantity describing the performance of an algorithm compared to some hypothesis. In other words, a regret of 0 or "no regret" signifies that, when learning over a large enough period of time, the average loss incurred by using the algorithm is just as large as if we had followed a certain hypothesis starting the beginning of the learning period. More specifically, the cumulative regret up to time T is defined as the difference between the losses incurred by the choices $\mathbf{x}_{1:T} = \mathbf{x}_1, \dots, \mathbf{x}_T$ up to time T and the loss which would have been incurred if an optimal $\mathbf{x}$ out of some set X had been chosen at all times t up to time T,

$$\frac{1}{T}\mathbf{Reg}_T = \frac{1}{T}\left(\sum_{t=1}^{T} l_t(\mathbf{x}_t, \mathbf{f}_t) - \inf_{\mathbf{x}\in X}\sum_{t=1}^{T} l_t(\mathbf{x}, \mathbf{f}_t)\right). \tag{1}$$

Let X be a pre-determined compact set. In other words, $\mathbf{x}_{1:T}$ is being compared to the competing hypothesis $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} \sum_{t=1}^{T} l_t(\mathbf{x}, \mathbf{f}_t)$ out of the compact set X if all information on Nature's pick $\mathbf{f}_{1:T} = \mathbf{f}_1, \dots, \mathbf{f}_T$ had been given. An algorithm then minimizes regret with respect to the hypothesis, if the regret formulation in (1) which compares the average loss incurred by algorithm to the average loss incurred if we had chosen $\mathbf{x}^*$ all along, goes to 0 as T increases. Specifically, in an online convex optimization problem the loss can be described as a convex function f_t solely depending on $\mathbf{x}$, i.e. $l_t(\mathbf{x}, \mathbf{f}_t) = f_t(\mathbf{x})$. Then the measurement of regret from equation (1) simplifies to

$$\mathbf{Reg}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}^*), \tag{2}$$

where we are using the same competing hypothesis as before, $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} \sum_{t=1}^{T} f_t(\mathbf{x})$.

2.1. Networked Online Convex Optimization

Consider a symmetric and connected network $\mathcal{G} = (V, \mathcal{E})$ with N nodes forming the vertex set $V = \{1, \dots, N\}$ and $|\mathcal{E}|$ edges in the edge set $\mathcal{E}$. Define the neighborhood of i as the set of nodes $n_i := \{j : (i, j) \in \mathcal{E}\}$ that share an edge with i. Each node in the network is associated with a sequence of cost functions $f_{i,t} : \mathbb{R}^J \to \mathbb{R}$ for all times $t \geq 0$. If a common variable $\mathbf{x}$ is played for all these functions the global network cost at time t is then given by

$$f_t(\mathbf{x}) = \sum_{i=1}^{N} f_{i,t}(\mathbf{x}). \tag{3}$$

Combining the definitions in (2) and (3) we can consider a coordinated game where all agents play a common variable $\mathbf{x}_t$ at time t. The accumulated regret associated with playing the sequence $\{\mathbf{x}_t\}_{t=1}^T$, as opposed to playing the optimal $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} \sum_{t=1}^T f_t(\mathbf{x})$ for all times t, can then be expressed as

$$\begin{aligned}\mathbf{Reg}_T^{\mathrm{C}} &= \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}^*) \\ &= \sum_{t=1}^T \sum_{i=1}^N f_{i,t}(\mathbf{x}_t) - \sum_{t=1}^T \sum_{i=1}^N f_{i,t}(\mathbf{x}^*). \end{aligned} \tag{4}$$

In this paper we are interested in distributed games in which each agent in the network plays his own variables $\mathbf{x}_{i,t}$ which are not necessarily identical to the variables $\mathbf{x}_{j,t}$ played by other agents $j \neq i$ in the same time slot. However, we are still focused in situations where each agent is interested in learning a play that is optimal with respect to the global cost in (3). Thus, we formulate a problem in which the regret of agent i is defined as

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

Except for the normalizing factor $1/N$ the regret formulations in (4) and (5) are identical. In particular, this means that the optimal play $\mathbf{x}^*$ is the same in both problems and that (5) corresponds to a problem in which agent i 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, the assumption here is that only the local functions $f_{i,t}$ are known to agent i.

By further considering the sum of all local regrets in (5) we can define a global version of networked regret as

$$\mathbf{Reg}_T := \sum_{i=1}^N \mathbf{Reg}_T^i = \frac{1}{N} \sum_{t=1}^T \sum_{i,j=1}^N f_{j,t}(\mathbf{x}_{i,t}) - \sum_{t=1}^T \sum_{i=1}^N f_{i,t}(\mathbf{x}^*), \tag{6}$$

where we used (5) and simplified terms to write the second equality.

In this paper we develop a variation of the saddle point algorithm of Arrow and Hurwicz [11] to find a strategy whose regrets 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 peers in their network neighborhood. The saddle point algorithm is presented in the following section after the discussion of an example and a pertinent remark.

Example 1 (Distributed recursive least squares) An example problem that admits the formulation in (6) is a distributed recursive least squares (RLS) problem. Suppose we want to estimate a signal $\mathbf{x} \in \mathbb{R}^J$ when agents collect observations $\mathbf{y}_{it} \in \mathbb{R}^K$ that relate to $\mathbf{x}$ according to the model $\mathbf{y}_{it} = \mathbf{H}_{i,t}\mathbf{x} + \mathbf{w}_{i,t}$, where the noise $\mathbf{w}_{i,t}$ is Gaussian independent and identically distributed across nodes and time. The optimal estimator $\mathbf{x}^*$ given the observations $\mathbf{y}_{i,t}$ for all i and t is the least mean squared error estimator $\mathbf{x}^* = \operatorname{argmin}_x \sum_{t=1}^T \sum_{i=1}^N \|\mathbf{H}_{i,t}\mathbf{x} - \mathbf{y}_{i,t}\|^2$. If the signals $\mathbf{y}_{i,t}$ are known for all nodes i and times t the optimal estimator $\mathbf{x}^*$ can be easily computed. In this paper we are interested in cases where the signal $\mathbf{y}_{i,t-1}$ is revealed at time $t-1$ to sensor i which then proceeds to causally estimate the signal $\mathbf{x}$ as $\mathbf{x}_{i,t}$ as a function of past observations $y_{i,u}$ for $u = 1, \ldots, t-1$ and information received from neighboring nodes in previous time slots. This is a distributed RLS problem because signals are revealed sequentially to agents of a network. Setting aside the issue of how to select $\mathbf{x}_{i,t}$, the regret in (6) is a measure of goodness for $\mathbf{x}_{i,t}$ with respect to a clairvoyant centralized estimator, and is formulated as

$$\mathbf{Reg}_T = \frac{1}{N} \sum_{t=1}^T \sum_{i,j=1}^N \|\mathbf{H}_{j,t}\mathbf{x}_{i,t} - \mathbf{y}_{j,t}\|^2 - \sum_{t=1}^T \sum_{i=1}^N \|\mathbf{H}_{i,t}\mathbf{x}^* - \mathbf{y}_{i,t}\|^2. \tag{7}$$

The regret $\mathbf{Reg}_T$ in (7) is measuring the mean squared error penalty that agent i is incurring by selecting $\mathbf{x}_{i,t}$ instead of the optimal estimator $\mathbf{x}^*$. In that sense it can be interpreted as the penalty for distributed causal operation with respect to centralized clairvoyant operation – the estimate $\mathbf{x}^*$ is centralized because it has access to the observations of all nodes and clairvoyant because it has access to the current observation $\mathbf{y}_{i,t}$. The algorithms developed in this paper are such that the regret $\mathbf{Reg}_T$ grows at a sub-linear rate – see Sections 3 and 4.

Remark 1 An alternative distributed regret formulation is to consider the aggregate cost $\sum_{i=1}^N f_{i,t}(\mathbf{x}_{i,t})$ incurred by each agent playing on its own local function. In such case we could define the regret by time T as

$$\mathbf{Reg}_T' = \sum_{t=1}^T \sum_{i=1}^N f_{i,t}(\mathbf{x}_{i,t}) - \sum_{t=1}^T \sum_{i=1}^N f_{i,t}(\mathbf{x}^*). \tag{8}$$

This formulation is of little interest because agents are independent of each other. Indeed, to reduce the regret in (8) it suffices to let agents learn strategies that are good with respect to their local costs $\sum_{t=1}^T f_{i,t}(\mathbf{x}_{i,t})$. A simple local gradient descent policy can achieve small regret with respect to the optimal *local* action $\tilde{\mathbf{x}}_i^* = \operatorname{argmin}_{\mathbf{x}} \sum_{t=1}^T f_{i,t}(\mathbf{x})$ [12] . This uncoordinated strategy is likely to result in negative regret in (8) since the variable $\mathbf{x}^*$ is chosen as common across all agents. The formulation in (5) and (6) is more appropriate as it is providing an incentive for agents to learn the cost functions of their peers. Lack of practical interest notwithstanding, the formulation in (8) plays an instrumental role in the determination of the regret bounds in Section 4

3. ARROW-HURWICZ SADDLE POINT ALGORITHM

For the presentation of a saddle point algorithm solving the optimization problem in (6), we will introduce a change in notation for simplicity. For the remainder of this paper, let $\mathbf{x}_t = \{\mathbf{x}_{i,t}\}_i$ denote a vector in which the actions of the nodes $i = 1, \ldots, N$ are stacked. Noting that the loss functions as defined in (5) are the same for every node, in the end the goal is to make decisions $\mathbf{x}_{i,t}$ which are the same for every node as well. Since the network $\mathcal{G}$ is assumed to be connected, this relationship can be described using the edge incidence matrix $\tilde{\mathbf{C}}$. Define a replicated version of the edge incidence matrix of the directed network as $\mathbf{C}$, where each 1, -1 and 0 from the edge incidence matrix $\tilde{\mathbf{C}}$ is replaced by the identity matrix $\mathbf{I}$, $-\mathbf{I}$ and the zero matrix $\mathbf{0}$ of size J respectively. Then equivalence between $\mathbf{x}_{i,t}$ and $\mathbf{x}_{j,t}$ for any nodes i and j can be rewritten using $\mathbf{C}$ as

$$\mathbf{C}\mathbf{x}_t = \mathbf{0} \qquad \forall t = 1, \ldots, T. \tag{9}$$

The edge incidence matrix $\mathbf{C}$ then has singular values $0 < \sigma_{\text{min}} \leq \cdots \leq \sigma_{\text{max}}$, where the smallest non-zero singular value reflects the connectivity of the network. While (9) describes the final goal, it makes no sense to enforce this relation for any time t. Instead consider penalizing the deviation from (9) using the time-dependent online Lagrangian at time t,

$$\mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) = \sum_{i=1}^N f_{i,t}(\mathbf{x}_{i,t}) + \boldsymbol{\lambda}_t^T \mathbf{C}\mathbf{x}_t, \tag{10}$$

by introducing dual variables $\boldsymbol{\lambda}_t$ for each time step t. The dual variables can be interpreted as a penalty term for relaxing the constraint in equation (9), so the components of the dual vector $\boldsymbol{\lambda}_t$ for each time t can be denoted by $\boldsymbol{\lambda}_t = \{\boldsymbol{\lambda}_{ij,t}\}_{i,j \in n_i}$, such that $\boldsymbol{\lambda}_{ij,t}$ is a vector of length J penalizing the disagreement of $\mathbf{x}_{i,t}$ and $\mathbf{x}_{j,t}$. Using the online Lagrangian from equation (10), the resulting Arrow-Hurwicz-algorithm then has the following form,

$$\mathbf{x}_{t+1} = \mathcal{P}_X[\mathbf{x}_t - \epsilon_t \nabla_{\mathbf{x}} \mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)] \tag{11}$$

$$\boldsymbol{\lambda}_{t+1} = \mathcal{P}_\Lambda[\boldsymbol{\lambda}_t + \epsilon_t \nabla_{\boldsymbol{\lambda}} \mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)], \tag{12}$$

where $\mathcal{P}_S$ denotes a projection onto set S and ϵ_t is the step size at time t. The primal update in (11) is reminiscent of the classical online gradient descent formulation, where the gradient of the objective is replaced by the gradient of the Lagrangian. The dual formulation from equation (12) is then a dual gradient ascent step which updates the dual variables depending on the constraint slack $\mathbf{C}\mathbf{x}_t$. At the end of this section, Remark 2 provides another intuition to motivate the method described by (11)-(12). Recent works on saddle point algorithms assume boundedness on the subgradients of the Lagrangian which require the primal and dual domains to be bounded. We have therefore introduced an additional projection to the Saddle Point Algorithm, and project the primal and dual variables onto compact sets X and Λ at each time step t, leading to the formulation in (11)-(12). In order to implement the Arrow-Hurwicz algorithm from (11)-(12) in a separable way, note that $\mathbf{x}_t = \{\mathbf{x}_{i,t}\}_i$ and $\boldsymbol{\lambda}_t = \{\boldsymbol{\lambda}_{ij,t}\}_{(i,j)}$ are stacked versions of $\mathbf{x}_{i,t}$ and $\boldsymbol{\lambda}_{ij,t}$. Therefore, we are interested in the separability of the gradients of the online Lagrangian with respect to $\mathbf{x}$ as in (11) and to $\boldsymbol{\lambda}$ as in (12). The gradient of the online Lagrangian with respect to the primal variable $\mathbf{x}_i$ of node i can be written as

$$\nabla_{\mathbf{x}_i}\mathcal{O}_t(\mathbf{x}_t,\boldsymbol{\lambda}_t) = \nabla_{\mathbf{x}_i} f_{i,t}(\mathbf{x}_{i,t}) + \sum_{j\in n_i}\boldsymbol{\lambda}_{ij,t}\mathbf{C}_{ij,t} \tag{13}$$

The computation of this gradient only depends on the local gradient of the loss function $f_{i,t}(\mathbf{x}_{i,t})$ and the dual variables for neighboring nodes j. Similarly, for each dual variable $\boldsymbol{\lambda}_{ij}$ for neighboring nodes i and j, the gradient of the online Lagrangian can be written as

$$\nabla_{\boldsymbol{\lambda}_{ij}}\mathcal{O}_t(\mathbf{x}_t,\boldsymbol{\lambda}_t) = \mathbf{C}_{ij,t}\mathbf{x}_t = \mathbf{x}_{i,t} - \mathbf{x}_{j,t}. \tag{14}$$

Therefore, to update each dual variable $\boldsymbol{\lambda}_{ij,t}$ at time t, it is only necessary to have access to the primal variables of neighboring nodes. Finally, to achieve separability, the projections onto sets X and Λ must be considered. When the sets $\{X_i\}_i$ and $\{\Lambda_{ij}\}_{(i,j)}$ are defined such that the resulting vectors $\mathbf{x}_{t+1}$ and $\boldsymbol{\lambda}_{t+1}$ are in the respective sets X and Λ, we can compute the primal and dual updates as follows,

$$\text{Primal: } \mathbf{x}_{i,t+1} = \mathcal{P}_{X_i}\Big[\mathbf{x}_{i,t} - \epsilon_t\Big(\nabla_{\mathbf{x}_i} f_{i,t}(\mathbf{x}_{i,t}) + \sum_{j\in n_i}\boldsymbol{\lambda}_{ij,t}\mathbf{C}_{ij,t}\Big)\Big] \tag{15}$$

$$\text{Dual: } \boldsymbol{\lambda}_{ij,t+1} = \mathcal{P}_{\Lambda_{ij}}\Big[\boldsymbol{\lambda}_{ij,t} + \epsilon_t\left(\mathbf{x}_{i,t} - \mathbf{x}_{j,t}\right)\Big]. \tag{16}$$

More precisely, at each time t, each pair of neighboring nodes (i, j) only needs to exchange dual variables $\boldsymbol{\lambda}_{ij,t}$ before the primal update step and primal variables $\mathbf{x}_{i,t}$ and $\mathbf{x}_{j,t}$ before the dual update.

Remark 2 Consider an equivalent problem to the regret minimization from equation (4). When treating the cost functions f_t as a random process, minimizing regret can be re-interpreted by minimizing the expected value of that process, $\min_{\mathbf{x}}\mathbb{E}\left[f_t(\mathbf{x})\right]$. More explicitly, when using the loss function $l_t(\mathbf{x},\mathbf{f})$ presented in (1), then Nature's play is the random process in question, such that the goal is to minimize the expected loss, i.e. $\min_{\mathbf{x}}\mathbb{E}_{\mathbf{f}}[l_t(\mathbf{x},\mathbf{f})]$. In order to make this problem separable over nodes, we look a the equivalent constrained formulation

$$\min_{\mathbf{x}}\mathbb{E}[f_t(\mathbf{x})], \quad \text{s.t. } \mathbf{C}\mathbf{x} = \mathbf{0}. \tag{17}$$

The Lagrangian of the problem,

$$\mathcal{L}_t(\mathbf{x}_t,\boldsymbol{\lambda}_t) = \mathbb{E}\left[f_t(\mathbf{x}_t)\right] + \boldsymbol{\lambda}_t^T\mathbf{C}\mathbf{x}_t, \tag{18}$$

can then be used to solve the optimization problem in (17) with the Arrow-Hurwicz algorithm of the form

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \epsilon_t\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t,\boldsymbol{\lambda}_t) \tag{19}$$

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \epsilon_t\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{x}_t,\boldsymbol{\lambda}_t). \tag{20}$$

However, the expectation in the Lagrangian in equation (18) cannot be separated, hence we replace the subgradients of the Lagrangian in equation (18) with the stochastic subgradients of the online Lagrangian at time t from equation (10). By adding projections onto X and Λ, the resulting saddle point algorithm is described by equations (11)-(12).

4. REGRET BOUNDS

To determine whether the saddle point algorithm described in (11)-(12) minimizes regret, we would like to show that the regret formulation of (6) goes to 0, which is equivalent to finding an upper bound on the algorithm's regret which goes to 0 with increasing time. Specifically, we compare the choices of the algorithm $\mathbf{x}_t$ and $\boldsymbol{\lambda}_t$ at time t with the optimal primal variable $\mathbf{x}^*$ and an arbitrary dual variable $\boldsymbol{\lambda}$ to find expressions for

$$\|\mathbf{x}_t - \mathbf{x}^*\|_2 \quad\text{and}\quad \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}\|_2, \tag{21}$$

which we can only bound in a time-average sense. Noting that the online Lagrangian evaluated for the primal choices $\mathbf{x}_{1:T}$ and for the optimal choice $\mathbf{x}^*$ is can be summed up to yield an expression which includes the uncoordinated regret formulation as (8), we use the distances of (21) to bound the sum of the online Lagrangians. The result for the global regret from (6) follows from the expression for (8). For the following results, we make some assumptions on the network, the primal and dual variables $\mathbf{x}_t$ and $\boldsymbol{\lambda}_t$, and the loss functions $f_t(\mathbf{x})$,

(A1) The network $\mathcal{G}$ is connected with diameter D.

(A2) The loss functions $f_{i,t}(\mathbf{x})$ are convex in $\mathbf{x}$ for any node i

$$f_{i,t}(\mathbf{x}) - f_{i,t}(\mathbf{y}) \le \nabla f_{i,t}(\mathbf{x})^T(\mathbf{x}-\mathbf{y}). \tag{22}$$

(A3) The gradients of the loss functions for any $\mathbf{x}$ is bounded by a constant L, i.e.

$$\|\nabla f_t(\mathbf{x})\|_2 \le L. \tag{23}$$

(A4) The loss functions $f_{i,t}(\mathbf{x})$ are Lipschitz continuous with modulus $l_{i,t}$, and $\ell = \max_{i,t} l_{i,t}$

$$\|f_{i,t}(\mathbf{x}) - f_{i,t}(\mathbf{y})\|_2 \le l_{i,t}\|\mathbf{x}-\mathbf{y}\|_2 \tag{24}$$

(A5) The primal variables $\mathbf{x}_{i,t}$ for any time t are projected into the set

$$X_i = \Big\{\mathbf{x}\in\mathbb{R}^J : \|\mathbf{x}\|_2 \le M/N\Big\} \tag{25}$$

(A6) The dual variables $\boldsymbol{\lambda}_{ij,t}$ for any time t are projected into the set

$$\Lambda = \left\{\boldsymbol{\lambda}\in\mathbb{R}^J : \|\boldsymbol{\lambda}\|_1 \le \max\{\frac{L}{\sqrt{|\mathcal{E}|}\sigma_{\min}}, DN\ell+1\}\right\} \tag{26}$$

Assumption (A1) is standard in distributed algorithms. Assumption (A2) follows from the problem formulation as an online convex optimization problem. Then the formulation of the online Lagrangian $\mathcal{O}_t(\mathbf{x},\boldsymbol{\lambda})$ from equation (10) is convex in the primal variable $\mathbf{x}$, such that

$$\mathcal{O}_t(\mathbf{x},\boldsymbol{\lambda}) - \mathcal{O}_t(\mathbf{y},\boldsymbol{\lambda}) \le \nabla_{\mathbf{x}}\mathcal{O}_t(\mathbf{x},\boldsymbol{\lambda})^T(\mathbf{x}-\mathbf{y}). \tag{27}$$

and concave in the dual variable $\boldsymbol{\lambda}$, i.e.

$$\mathcal{O}_t(\mathbf{x},\boldsymbol{\lambda}) - \mathcal{O}_t(\mathbf{x},\boldsymbol{\mu}) \ge \nabla_{\boldsymbol{\lambda}}\mathcal{O}_t(\mathbf{x},\boldsymbol{\lambda})^T(\boldsymbol{\mu}-\boldsymbol{\lambda}). \tag{28}$$

Assumptions (A3)-(A6) are mild assumptions typical in the analysis of saddle point algorithms. The projections into the sets $\{X_i\}_i$ and $\{\Lambda_{ij}\}_{(i,j)}$ in assumptions (A5) and (A6) are constructed such that the optimal primal $\mathbf{x}^*$ and dual variables $\boldsymbol{\lambda}^*$ can be bounded with the same respective constants. For the stacked primal variable $\mathbf{x}_t$, (A5) implies that it can be bounded by $\|\mathbf{x}_t\|_2 \le M$ at any time t. Using the identity $\|\boldsymbol{\lambda}_t\|_1 \le \sqrt{|\mathcal{E}|J}\,\|\boldsymbol{\lambda}_t\|_2$ relating the 1-norm of the vector $\boldsymbol{\lambda}_t$ to its 2-norm, we can bound the dual variable by $\|\boldsymbol{\lambda}_t\|_2 \le \sigma_{\max}\max\{L/\sigma_{\min}, \sqrt{|\mathcal{E}|}(DN\ell+1)$, such that $\boldsymbol{\lambda}_t$ is delimited by the smoothness of the cost function and the connectivity of the

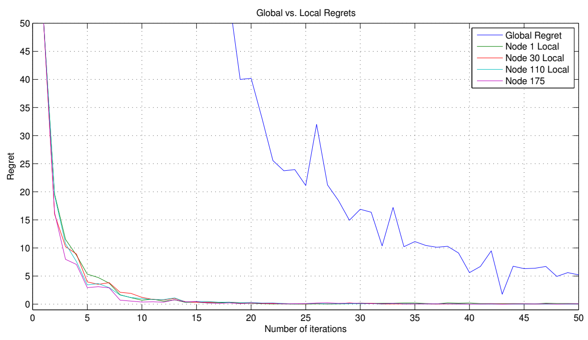

Fig. 1. Normalized aggregate regret $\mathbf{Reg}_T/NT$ and individual regret $\mathbf{Reg}_T^i/T$ for representative nodes. Observe that $\mathbf{Reg}_T/NT$ vanishes consistent with the result in Theorem 2. Individual regrets also vanish although that is not theoretically guaranteed.

network. With these bounds, the gradients of the online Lagrangians can also be bounded. For the gradient with respect to the primal variable $\mathbf{x}$, using the Cauchy-Schwarz and triangle inequalities yields

$$\begin{aligned}\|\nabla_{\mathbf{x}}\mathcal{O}_t(\mathbf{x}_t,\boldsymbol{\lambda}_t)\|_2 &= \|\nabla f_t(\mathbf{x}_t)+\mathbf{C}^T\boldsymbol{\lambda}_t\|_2\\ &\le \|\nabla f_t(\mathbf{x}_t)\|_2+\|\mathbf{C}^T\|_2\|\boldsymbol{\lambda}_t\|_2\\ &\le L+\sigma_{\max}\max\{\frac{L}{\sigma_{\min}},\sqrt{|\mathcal{E}|}(DN\ell+1)\}:=L_{\mathbf{x}}\end{aligned} \tag{29}$$

where the last inequality above uses the fact that the singular values of $\mathbf{C}$ and $\mathbf{C}^T$ are the same. Similarly, for the gradient with respect to the dual variable $\boldsymbol{\lambda}$, we can similarly write

$$\begin{aligned}\|\nabla_{\boldsymbol{\lambda}}\mathcal{O}_t(\mathbf{x}_t,\boldsymbol{\lambda}_t)\|_2 &= \|\mathbf{C}\mathbf{x}_t\|_2\le\|\mathbf{C}\|_2\|\mathbf{x}_t\|_2\\ &\le \sigma_{\max}M:=L_{\boldsymbol{\lambda}}\end{aligned} \tag{30}$$

using the same assumptions. Then the main results of this paper concern the global regret from (6) as well as the uncoordinated regret from (8) of the algorithm from equations (11)-(12) which we bound with a term that goes 0 as the final learning time T goes to infinity.

Theorem 1 *Let T be the final learning time. Let $\mathbf{x}_t=\{\mathbf{x}_{i,t}\}_i$ be a vector of the learning network's choice as a result from the algorithm in equations* (11)-(12)*, and let $\mathbf{x}^*$ be the optimal choice if the loss functions $f_{1:T}(\mathbf{x})$ were given for all times and all nodes. Let the dual variables be initialized at $\boldsymbol{\lambda}_0=\mathbf{0}$. Assume the step size is constant with $\epsilon_t=1/\sqrt{T}$. If assumptions (A1) to (A6) hold, then the uncoordinated regret from choosing $\mathbf{x}_t$ can be bounded by*

$$\mathbf{Reg}_T'\le\frac{\sqrt{T}}{2}\left(\|\mathbf{x}_1-\mathbf{x}^*\|_2^2+L_{\mathbf{x}}^2+L_{\boldsymbol{\lambda}}^2\right)=O(\sqrt{T}). \tag{31}$$

Proof: See [13] □

From Theorem 1, we can use a variation of the triangle inequality to yield a result for the global regret as formulated in (6). The result can be stated as follows.

Theorem 2 *Let T be the final learning time. Let $\mathbf{x}_t=\{\mathbf{x}_{i,t}\}_i$ be a vector of the learning network's choice as a result from the algorithm in equations* (11)-(12)*, and let $\mathbf{x}^*$ be the optimal choice if the loss functions $f_{1:T}(\mathbf{x})$ were given for all times and all nodes. Let the dual variables be initialized at $\boldsymbol{\lambda}_0=\mathbf{0}$. Assume the step size is constant with $\epsilon_t=1/\sqrt{T}$. If assumptions (A1) to (A6) hold, then the total regret from choosing $\mathbf{x}_t$ can be bounded by*

$$\mathbf{Reg}_T\le\frac{\sqrt{T}}{2}\left(\|\mathbf{x}_1-\mathbf{x}^*\|_2^2+|\mathcal{E}|(DN\ell+1)^2+L_{\mathbf{x}}^2+L_{\boldsymbol{\lambda}}^2\right)=O(\sqrt{T}). \tag{32}$$

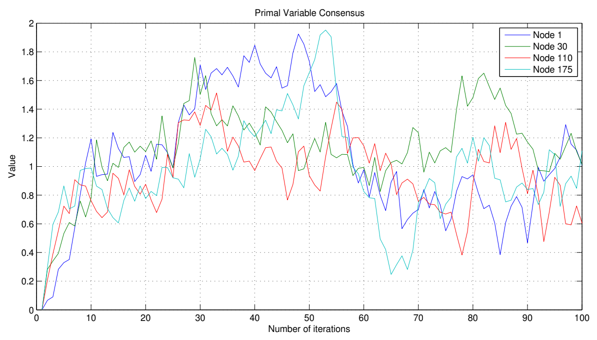

Fig. 2. Evolution of the variables $\mathbf{x}_{i,t}$ for a set of representative nodes. All nodes converge to the same value as they all learn the individual information available to all peers.

Proof: See [13] □

As is standard for online convex optimization algorithms, we establish a bound of order $O(\sqrt{T})$ when the step size is chosen to be $\epsilon_t=1/\sqrt{T}$. The rate at which regret vanishes depends on the choice of the final learning time T. The constants in the bound in (31) and (32) also depend on the size of $\mathbf{x}_0$ which can be bounded by M. Furthermore, a large variability of the objective $f_t(\mathbf{x}_t)$ leads to a larger bound, and the bounds on the online Lagrangian in (29) and (30) show that a higher connectivity of the network, reflected in a small ratio $\sigma_{\max}/\sigma_{\min}$, leads to a smaller regret bound. For the global regret bound in (32) specifically, size and the diameter of the network play an additional role where a large network and a larger amount of hops between nodes, i.e. a large diameter, decrease the algorithm convergence rate.

5. SIMULATION RESULTS

For the distributed RLS regret minimization problem in Example 1, the primal update of the saddle point algorithm takes the form

$$\mathbf{x}_{i,t+1}=\mathcal{P}_{X_i}\Big[\mathbf{x}_{i,t}-\epsilon_t\Big(2\mathbf{H}_{i,t}^T\mathbf{H}_{i,t}\mathbf{x}_{i,t}-2\mathbf{H}_{i,t}^T\mathbf{y}_{i,t}+\sum_{j\in n_i}\boldsymbol{\lambda}_{ij,t}\mathbf{C}_{ij,t}\Big)\Big] \tag{33}$$

while the dual update remains (16). We implement the iteration (33) - (16) for a network with $N=200$ nodes and edges randomly generated so that nodes are connected with probability $1/5$. The matrix $\mathbf{H}_{i,t}\in\mathbb{R}^{11\times6}$ is generated from taking a vector $u\in\mathbb{R}^{11}$ with $u_k=10^{-k}$ and stacking column-wise increasing powers of u, with the jth column of $\mathbf{H}_{i,t}$ given by u^j for $j=1,\dots,6$. We set the coefficients $\mathbf{y}_{i,t}=\mathbf{H}_{i,t}\mathbf{x}+\mathbf{w}_{i,t}$ where $\mathbf{x}=\mathbf{1}$ and $\mathbf{w}_{i,t}$ sampled from a zero-mean, $\sigma^2=4$ normal distribution. We run (33) - (16) for a total of $T=2\times10^3$ iterations.

Normalized aggregate regret $\mathbf{Reg}_T/NT$ as well as individual regret $\mathbf{Reg}_T^i/T$ for representative nodes are shown in Fig. 1. $\mathbf{Reg}_T/NT$ vanishes consistent with the result in Theorem 2. While we don't have theoretical guarantees on individual regret our numerical experiments indicate that individual normalized regrets $\mathbf{Reg}_T^i/T$ also vanish. Fig. 2 shows the evolution of the variables $\mathbf{x}_{i,t}$ for a set of representative nodes. These variables converge towards a common value, which is as expected since each individual node learns all the information available throughout the network.

6. REFERENCES

[1] C. B. Do, Q. V. Le, and C. Foo, “Proximal regularization for online and batch learning.,” in *ICML*, A.P. Danyluk, L. Bottou, and M.L. Littman, Eds. 2009, vol. 382 of *ACM International Conference Proceeding Series*, p. 33, ACM.

[2] A. Nedic and A. Ozdaglar, “Approximate primal solutions and rate analysis for dual subgradient methods,,” *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1757–1780, 2008.

[3] 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)*, Atlanta, GA, 2013, vol. 28, pp. 392–400, JMLR Workshop and Conference Proceedings.

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

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

[6] S. Shalev-Shwartz, “Online learning and online convex optimization,” *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, Feb. 2012.

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

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

[9] A. Nedic and A. Ozdaglar, “Subgradient methods for saddle-point problems,” *Journal of Optimization Theory and Applications*, pp. 205–228, 2009.

[10] S. Boyd and L. Vanderberghe, *Convex Programming*, Wiley, New York, NY, 2004.

[11] K.J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in linear and non-linear programming*, 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.

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

[13] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, “A saddle point algorithm for networked online convex optimization,” *in preparation*, 2013.