A SADDLE POINT ALGORITHM FOR NETWORKED ONLINE CONVEX OPTIMIZATION

Alec Koppel, Felicia Y. Jakubiec and Alejandro Ribeiro Contact E-mail: {akoppel, life, aribeiro}@seas.upenn.edu

Penn
University of Pennsylvania

Online Convex Optimization

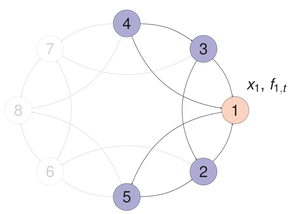
- Repeated prediction game over a convex set $X \subset \mathbb{R}^J$
- At the t_{th} round,
  - $\Rightarrow$ Predictor plays $x_t \in \mathbb{R}^J$
  - $\Rightarrow$ Nature reveals convex loss function $f_t : \mathbb{R}^J \to \mathbb{R}$
  - $\Rightarrow$ Predictor suffers loss $f_t(x_t)$
- Examples
  - $\Rightarrow$ Online signal detection (Recursive Least Squares)
  - $\Rightarrow$ Detection of malicious users in Local Area Networks
- For T rounds of the game, define Regret of online algorithm as

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

  - $\Rightarrow$ $x^* = \text{argmin}_{x \in X} \sum_{t=1}^{T} f_t(\mathbf{x})$
  - $\Rightarrow$ x^* is batch learner: all loss functions revealed in advance
- Goal: to have sublinear Regret growth with the total number of iterations

Networked Sequential Prediction and Coordinated Regret

- Network $\mathcal{G} = (V, \mathcal{E})$, $|V| = N$.
- $N_i = \{j : (j, i) \in E\}$
  - $\Rightarrow$ Neighborhood of node i
- Repeated game at each node i
  - $\Rightarrow$ action $\mathbf{x}_{i,t}$ and loss $f_{i,t}$
  - $\Rightarrow$ Use information at N_i
- Coordinated game at time t:
  - $\Rightarrow$ All agents play common $\mathbf{x}_t$



- Coordinated Regret

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

- We want to solve distributed games:
  - $\Rightarrow$ $\mathbf{x}_{i,t}$ action of node i at time t
  - $\Rightarrow$ $\mathbf{x}_{j,t}$ chosen independently for $j \neq i$
- Want nodes to select actions independently and minimize $\mathbf{Reg}_T^{\mathrm{C}}$
- Hence we focus on minimizing $\sum_{i=1}^{N} f_{i,t}(\mathbf{x})$ with only local information

Uncoordinated, Local, and Global Networked Regret

- When nodes can autonomously select actions, multiple possible metrics:
- Uncoordinated Regret

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

- $\mathbf{Reg}_T'$ Measures each node's online learner vs. the batch solution
- Poor metric for distributed online learning
  - $\Rightarrow$ No incorporation of node discrepancy
- **Local Regret for node i** of distributed online algorithm

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

- $\mathbf{x}^* = \text{argmin}_{\mathbf{x}} \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\mathbf{x})$ is the global batch learner
- **Global Networked Regret**

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

- Network average of local regrets
  - $\Rightarrow$ Measures Distance from batch learner and network disagreement

Lagrangian Relaxation and Set Projections

- At each t we approximately enforce $\mathbf{x}_{i,t} = \mathbf{x}_{j,t}$ for $j \in N_i$, or $\mathbf{C}\mathbf{x}_t = \mathbf{0}$.
  - $\Rightarrow$ $\mathbf{C}$ is node-edge incidence matrix of $\mathcal{G}$.
  - $\Rightarrow$ $\mathbf{x}_t$ is stacked version of $\mathbf{x}_{i,t}$.
- Online Lagrangian relaxation for sequential learning problem:

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

- Convex/concave function in the primal/dual variables, respectively
- Need set projections for bounded Lagrangian primal/dual subgradients
  - $\Rightarrow$ project primal variable $\mathbf{x}_{i,t}$ onto the set

$$X_i = \left\{\mathbf{x} \in \mathbb{R}^J : \|\mathbf{x}\|_2 \leq M/N\right\}$$

  - $\Rightarrow$ project dual variable $\boldsymbol{\lambda}_{ij,t}$ onto the set

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

  - $\Rightarrow$ Norm ball around primal optimum and worst-case constraint slack

Saddle Point Algorithm

- Primal Lagrangian subgradient descent
  - $\Rightarrow$ Minimize loss
  - $\Rightarrow$ Dual correction term to approximately enforce local agreement
- Dual Lagrangian subgradient ascent
  - $\Rightarrow$ Penalize local discrepancy
- **Algorithm formulation:**

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

- **Local coordinate formulation:**

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

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

- Initialize algorithm with $\boldsymbol{\lambda}_1 = \mathbf{0}$ for $\boldsymbol{\lambda}_t$ to remain in the image of $\mathbf{C}$
  - $\Rightarrow$ Required for bounded dual subgradients and constraint slacks
- Dual update stays the same across different applications.
- Approximately iterate towards the saddle point of the problem

$$\min_{x_i} \sum_{t=1}^{T}\sum_{i=1}^{N} f_{i,t}(\mathbf{x}_{i,t}) \qquad \text{s.t.} \quad \mathbf{C}\mathbf{x}_t = \mathbf{0}.$$

Connection with Statistical Learning

- Treat the cost functions f_t as a random process
- Re-interpret regret minimization as minimizing $\min_{\mathbf{x}} \mathbb{E}[f_t(\mathbf{x})]$
- When f_t is node separable, the equivalent constrained formulation becomes

$$\min_{\mathbf{x} \in X} \ \mathbb{E}[f_t(\mathbf{x})] \qquad \text{s.t.} \quad \mathbf{C}\mathbf{x} = \mathbf{0}$$

- Lagrangian of this problem:

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

- Compute primal and dual Lagrangian subgradients to obtain

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

- This algorithm is not distributable over network (expectation not separable)
- With the following modifications we obtain the saddle point algorithm . . .
  - $\Rightarrow$ Replace $\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ subgradients with online subgradients of $\mathcal{O}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)$.
  - $\Rightarrow$ Add set projections to ensure bounded stochastic subgradients.

Networked Regret Bounds

Lemma: *With step-size $\epsilon_t = 1/\sqrt{T}$, the Uncoordinated Regret with a constraint slack of the saddle point iterates is bounded above at a rate of $O(\sqrt{T})$:*

$$\sum_{t=1}^{T}\sum_{i=1}^{N}\left(f_{i,t}(\mathbf{x}_{i,t}) - f_{i,t}(\mathbf{x}^*)\right) + \sum_{t=1}^{T}\boldsymbol{\lambda}^{\mathsf{T}}\mathbf{C}\mathbf{x}_t \leq \frac{\sqrt{T}}{2}\left(\|\mathbf{x}_1 - \mathbf{x}^*\|_2^2 + \|\boldsymbol{\lambda}\|_2^2 + L_{\mathbf{x}}^2 + L_{\boldsymbol{\lambda}}^2\right).$$

Theorem: *With step-size $\epsilon_t = 1/\sqrt{T}$, the saddle point iterates achieve an $O(\sqrt{T})$ Global Networked Regret bound:*

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

Regularity and Network Assumptions

- The network $\mathcal{G}$ is connected with diameter D.
- The loss functions $f_{i,t}(\mathbf{x})$ are convex in $\mathbf{x}$ for any node i

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

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

$$\|\nabla f_i(\mathbf{x})\|_2 \leq L.$$

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

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

Distributed Recursive Least Squares

- We want to estimate a signal $\mathbf{x} \in \mathbb{R}^J$ from observations $\mathbf{y}_{i,t} \in \mathbb{R}^K$
  - $\Rightarrow$ Relationship between observations and estimate: $\mathbf{y}_{i,t} = \mathbf{H}_{i,t}\mathbf{x} + \mathbf{w}_{i,t}$
  - $\Rightarrow$ $\mathbf{w}_{i,t}$ is Gaussian and i.i.d across time and node
  - $\Rightarrow$ Optimal estimator is $\mathbf{x}^* = \text{argmin}_x \sum_{t=1}^{T}\sum_{i=1}^{N} \|\mathbf{H}_{i,t}\mathbf{x} - \mathbf{y}_{i,t}\|^2$
- Global Networked Regret:

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

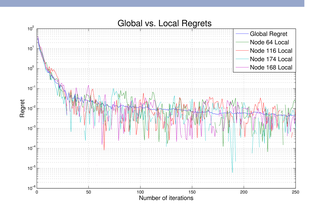
- Saddle Point Algorithm for Distributed Recursive Least Squares (DRLS):

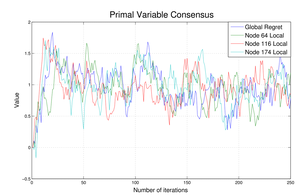
$$\mathbf{x}_{i,t+1} = \mathcal{P}_{X_i}\Big[\mathbf{x}_{i,t} - \epsilon_t\Big(2\mathbf{H}_{i,t}^T\mathbf{H}_{i,t}\mathbf{x}_{i,t} - 2\mathbf{H}_{i,t}^T\mathbf{y}_{i,t} + \sum_{j \in n_i}\boldsymbol{\lambda}_{ij,t}\mathbf{C}_{ij,t}\Big)\Big]$$

Networked Online Signal Estimation

- $N = 200$, $T = 2 \times 10^3$
- Random uniformly generated edges
  - $\Rightarrow$ Nodes connected w.p. 0.2
- $\mathbf{Reg}_T/NT$ vs. local regrets $\mathbf{Reg}_T^i/T$
  - $\Rightarrow$ Random sample of nodes
- $\mathbf{H}_{i,t} \in \mathbb{R}^{11 \times 6}$ generated by
  - $\Rightarrow$ $u \in \mathbb{R}^{11}$, $u_k = 10^{-k}$
  - $\Rightarrow$ Stack column-wise u^i
- $\mathbf{x} = \mathbf{1} \in \mathbb{R}^{11}$
- $\mathbf{w}_{i,t} \sim \mathcal{N}(0, \sigma^2 = 4)$
- Global Networked Regret
  - $\Rightarrow$ grows sublinearly with T.
  - $\Rightarrow$ reflects local regret average.

Primal consensus achieved.





Distributed Online Support Vector Classification

- Train binary support vector classifiers online
  - $\Rightarrow$ Each node observes different data
- $\mathbf{z}_{i,t} \in \mathbb{R}^J$ $\Rightarrow$ feature vector observed at node i, time t
- $\mathbf{y}_{i,t} \in \{-1, 1\}$ $\Rightarrow$ binary label variable
- ζ $\Rightarrow$ regularization parameter, trades off bias and complexity
- $\max(0, 1 - \mathbf{y}_t \cdot \mathbf{x}^T\mathbf{z}_t)$ $\Rightarrow$ hinge loss, penalizes misclassified vectors
- Global batch classifier solution to SVM optimization problem (with $N = 1$):

$$\min_{\mathbf{x} \in X} \frac{\zeta}{2}\|\mathbf{x}\|_2^2 + \frac{1}{d}\sum_{t=1}^{d}\max(0, 1 - \mathbf{y}_t \cdot \mathbf{x}^T\mathbf{z}_t)$$

- Distributed online formulation of training SVM classifier:

$$\min_{\mathbf{x}_i \in X} \quad \frac{\zeta}{2}\sum_{t=1}^{T}\sum_{i=1}^{N}\|\mathbf{x}_i\|_2^2 + \sum_{t=1}^{T}\frac{1}{t}\sum_{k=1}^{t}\sum_{i=1}^{N}\max(0, 1 - \mathbf{y}_{i,k} \cdot \mathbf{x}_i^{\mathsf{T}}\mathbf{z}_{i,k})$$
$$\text{s.t.} \quad \mathbf{C}\mathbf{x} = \mathbf{0}$$

- t feature vectors received at time t for node i
- Primal set projection onto $X = \{\mathbf{x}_i \in \mathbb{R}^J : \|\mathbf{x}_i\|_2 \leq \zeta\}$
- Loss functions measure node i's classifier performance on other nodes' data

$$f_{j,t}(\mathbf{x}_{i,t}) = \frac{\zeta}{2}\|\mathbf{x}_{i,t}\|_2^2 + \frac{1}{t}\sum_{k=1}^{t}\max\{0, 1 - \mathbf{y}_{j,k} \cdot \mathbf{x}_{i,t}^{\mathsf{T}}\mathbf{z}_{j,k}\}$$

- Saddle Point Algorithm: Distributed Online SVM Classification

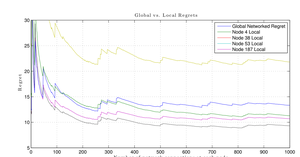
$$\mathbf{x}_{i,t+1} = \mathcal{P}_{X_i}\Big[\mathbf{x}_{i,t} - \epsilon_t\Big(\zeta\mathbf{x}_{i,t} + \frac{1}{t}\sum_{\{s:\mathbf{y}_{i,s}\mathbf{x}_{i,t}^{\mathsf{T}}\mathbf{z}_{i,s}<1\}}\mathbf{y}_s\mathbf{z}_s + \sum_{j \in n_i}\boldsymbol{\lambda}_{ij,t}\mathbf{C}_{ij,t}\Big)\Big]$$

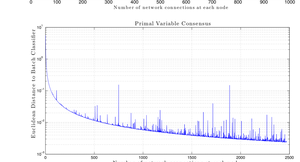
Computer Network Security

- Track connections in computer networks $\Rightarrow$ detect network attacks
- Record network protocol information such as
  - $\Rightarrow$ User bit-rates
  - $\Rightarrow$ Number of failed logins
  - $\Rightarrow$ Protocol types
- Nodes don't share local user IP information in respective networks
  - $\Rightarrow$ Only share attacker profile each node learns via local users
- Computer Network protocol
  - $\Rightarrow$ Identify malicious users
- $N = 200$ node network
- Random uniformly generated edges
  - $\Rightarrow$ Nodes connected w. p. 0.2
- KDDCup 99 data set:
  - $\Rightarrow$ $NT \approx 5 \times 10^5$ feature vectors
  - $\Rightarrow$ $J = 41$ features.
- $O(\sqrt{T})$ Regret bound rate
  - $\Rightarrow$ nodes learn attacker profile
- Euclidean distance to batch learner:

$$\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{\|x_{i,t} - x^*\|_2}{\|x^*\|}$$

  - $\Rightarrow$ Primal consensus achieved





Conclusions

- Networked regret minimization $\Rightarrow$ distributed learning goal
- Algorithm to solve networked online convex optimization problems
- Nodes make autonomous learning decisions $\Rightarrow$ use local information
  - $\Rightarrow$ Online Lagrangian to penalize node disagreement
  - $\Rightarrow$ Global Networked Regret grows at rate of $O(\sqrt{T})$.
- We demonstrate algorithm performance in DRLS and DSVM problems