Conservative Multi-agent Online Kernel Learning in Heterogeneous Networks

Hrusikesha Pradhan [†], Amrit Singh Bedi [§], Alec Koppel[§], Ketan Rajawat [†]

[†] Dept. of EE, India Institute of Technology Kanpur,

[§] CISD, U.S. Army Research Laboratory

IEEE Asilomar Conference on Signals, Systems, and Computers

Nov. 2, 2020

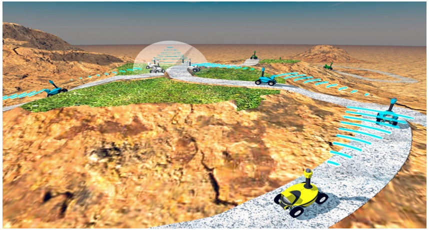
The State of Distributed Learning

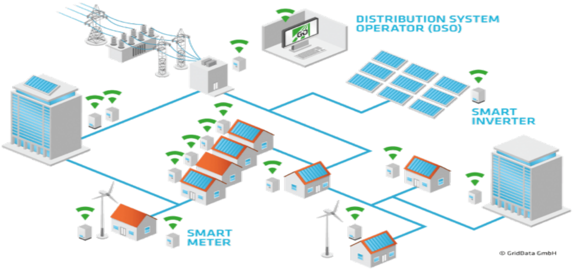
Consensus

Linear Statistical Models

Offline

Non-linear Heterogeneous Networked Learning





Existing methods don't apply

Consensus

Linear Statistical Models

Supervised Learning

$$\mathbf{y}_i = \mathbf{w}_i^\top \mathbf{x}_i$$

Offline

Existing methods don't apply

Consensus

Non-linear Statistical Models → Supervised Learning

$$y = f(\mathbf{x})$$

Offline

Existing methods don't apply

Consensus

$\mathbf{w}_i = \mathbf{w}_j \quad \text{for all} \quad j \in n_i$

Non-linear Statistical Models

Supervised Learning

$y = f(\mathbf{x})$

Offline

Existing methods don't apply

Proximity

$h_{ij}(f_i(\mathbf{x}_i), f_j(\mathbf{x}_i)) \leq \gamma_{ij}, \text{ for all } j \in n_i$

Non-linear Statistical Models

Supervised Learning

$y = f(\mathbf{x})$

Offline

Existing methods don't apply

Proximity

$$h_{ij}(f_i(\mathbf{x}_i), f_j(\mathbf{x}_i)) \leq \gamma_{ij}, \quad \text{for all} \quad j \in n_i$$

Non-linear Statistical Models

Supervised Learning

$$y = f(\mathbf{x})$$

Offline

$$\mathbf{w}^* := \underset{\mathbf{w} \in \mathbb{R}^{p|\mathcal{V}|}}{\text{argmin}} \frac{1}{N} \sum_{i=1}^{|\mathcal{V}|} \sum_{n=1}^{N} \ell_i(\mathbf{w}_i^\top \mathbf{x}_{i,n}, \mathbf{y}_{i,n})$$

Proximity

$$h_{ij}(f_i(\mathbf{x}_i), f_j(\mathbf{x}_i)) \leq \gamma_{ij}, \text{ for all } j \in n_i$$

Non-linear Statistical Models

Supervised Learning

$$y = f(\mathbf{x})$$

Online

$$f^\star = \underset{\{f_i\}\in\mathcal{H}}{\operatorname{argmin}} \sum_{i\in\mathcal{V}} \mathbb{E}_{\mathbf{x}_i, y_i}\left[\ell_i(f_i(\mathbf{x}_i), y_i)\right]$$

Contribution

- Proposed a non-linear function learning algorithm considering
  - $\Rightarrow$ Online settings
  - $\Rightarrow$ Network heterogeneity
- Non-asymptotic bound on the model complexity of the algorithm
- Characterizing the optimality gap in terms of
  - $\Rightarrow$ Model complexity
  - $\Rightarrow$ Number of iterations
- Null constraint violation (Conservative approach)

Contribution

Distributed Online Learning	Linear	Nonlinear
Homogeneous		Koppel ,
Heterogeneous	Sayed, Chen, Nassif, Lee, Sadler	**Our Work**

Haoran Sun, Mingyi Hong, Nikos D. Sidiropoulos

A. Nedic and A. Ozdaglar, Distributed subgradient methods for multiagent optimization, IEEE Transactions on Automatic Control

Mokhtari, Aryan, and Alejandro Ribeiro. "DSA: Decentralized double stochastic averaging gradient algorithm." JMLR

Sirb, B., Ye, X. (2018). Decentralized consensus algorithm with delayed and stochastic gradients. SIAM Journal on Optimization.

Chen, T., Ling, Q., Giannakis, G. B. (2017). An online convex optimization approach to proactive network resource allocation. IEEE Transactions on Signal Processing, 65(24), 6350-6364.

Sun Haoran, and Mingyi Hong. "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms." arXiv preprint arXiv:1804.02729 (2018)

Sun Haoran, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nikos D. Sidiropoulos. "Learning to optimize: Training deep neural networks for wireless resource management." In SPAWC, 2017 IEEE 18th International Workshop on, pp. 1-6. IEEE, 2017.

Chen, J., Richard, C. and Sayed, A. H. (2014). Multitask diffusion adaptation over networks. IEEE Transactions on Signal Processing

Nassif, R., Richard, C., Ferrari, A. and Sayed, A. H. Distributed learning over multitask networks with linearly related tasks.

Lee, S., Zavlanos, M. M. (2017). On the sublinear regret of distributed primal-dual algorithms for online constrained optimization.

Works on improving Constraint Violation rate

Journal of Machine Learning Research 13 (2012) 2503-2528 Submitted 8/11; Revised 3/12; Published 9/12

Trading Regret for Efficiency: Online Convex Optimization with Long Term Constraints

Mehrdad Mahdavi MAHDAVIM@CSE.MSU.EDU
Rong Jin RONGJIN@CSE.MSU.EDU
Tianbao Yang YANGTIA1@MSU.EDU
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI, 48824, USA

Editor: Shie Mannor

Abstract

In this paper we propose efficient algorithms for solving constrained online convex optimization problems. Our motivation stems from the observation that most algorithms proposed for online convex optimization require a projection onto the convex set $\mathcal{K}$ from which the decisions are made.

Works on improving Constraint Violation rate

Journal of Machine Learning Research 13 (2012) 2503-2528 Submitted 8/11; Revised 3/12; Published 9/12

Trading Regret for Efficiency:
Online Convex Optimization with Long Term Constraints

Mehrdad Mahdavi
Rong Jin
Tianbao Yang
Department of Computer Science a
Michigan State University
East Lansing, MI, 48824, USA

Editor: Shie Mannor

In this paper we propose effic
problems. Our motivation ste
convex optimization require a p

IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, VOL. 5, NO. 3, SEPTEMBER 2019 479

Asynchronous Online Learning in Multi-Agent Systems With Proximity Constraints

Amrit Singh Bedi, *Student Member, IEEE*, Alec Koppel, *Member, IEEE*, and Ketan Rajawat, *Member, IEEE*

***Abstract*—We consider the problem of distributed learning from sequential data via online convex optimization. A multi-agent system is considered where each agent has a private objective but is willing to cooperate in order to minimize the network cost, which is the sum of local cost functions. Different from the classical distributed settings, where the agents coordinate through the use of consensus constraints, we allow the neighboring agent actions to be related via a non-linear proximity function. A decentralized saddle point algorithm is proposed that is capable of handling gradient delays arising from computational issues. The proposed online asynchronous algorithm is analyzed under adversarial settings by developing bounds on the regret of $\mathcal{O}(\sqrt{T})$, which measures the cumulative loss incurred by the online algorithm against a clairvoyant, and network discrepancy of $\mathcal{O}(T^{3/4})$, which measures the cumulative constraint violation or agent disagreement. By allowing the agents to utilize stale gradient information, the proposed algorithm embraces the nuances of distributed learning and serves to be the first**

power budget judiciously but also refrain from communicating unnecessarily [4], [5]. Most multi-agent systems are also heterogeneous and have disparate sleep schedules that do not allow continuous data exchange at high rates. The modern approach towards handling networks with such reticent and heterogeneous agents is to explicitly design distributed algorithms that can tolerate errors and delays while allowing the agents to 'fall behind' or 'catch up' intermittently.

This work considers the problem of distributed online learning with constraints. Keeping the vagaries of distributed operation in mind, the goal is to design asynchronous and flexible optimization algorithms that can learn from the sequential data. We adopt the perspective of online convex optimization [6], where at each time slot, the learner selects an action (defining a para-

Works on improving Constraint Violation rate

Journal of Machine Learning Research 13 (2012) 2503-2528 Submitted 8/11; Revised 3/12; Published 9/12

Online Conv

Mehrdad Mahdavi
Rong Jin
Tianbao Yang
Department of Computer Scie
Michigan State University
East Lansing, MI, 48824, USA

Editor: Shie Mannor

In this paper we propose
problems. Our motivati
convex optimization requ

Conditional Value at Risk Sensitive Optimization via Subgradient Methods

Avinash N. Madavan Subhonmesh Bose*

April 16, 2020

Abstract

We study a first-order primal-dual subgradient method to optimize risk-constrained risk-penalized optimization problems, where risk is modeled via the popular conditional value at risk (CVaR) measure. The algorithm processes independent and identically distributed samples from the underlying uncertainty in an online fashion, and produces an $\eta/\sqrt{K}$-approximately feasible and $\eta/\sqrt{K}$-approximately optimal point within K iterations with constant step-size, where η increases with tunable risk-parameters of CVaR. We find optimized step sizes using our bounds and precisely characterize the computational cost of risk aversion as revealed by the growth in η. Our proposed algorithm makes a simple modification to a typical primal-dual

479

Agent

, *Member, IEEE*

tive loss incurred by the online algorithm against a clairvoyant, and network discrepancy of $\mathcal{O}(T^{3/4})$, which measures the cumulative constraint violation or agent disagreement. By allowing the agents to utilize stale gradient information, the proposed algorithm embraces the nuances of distributed learning and serves to be the first

from communicating
ystems are also het-
ules that do not allow
he modern approach
nt and heterogeneous
gorithms that can tol-
gents to 'fall behind'

buted online learning
distributed operation
in mind, the goal is to design asynchronous and flexible optimization algorithms that can learn from the sequential data. We adopt the perspective of online convex optimization [6], where at each time slot, the learner selects an action (defining a para-

Comparison with previous work

- Here we have considered a different approach to solve the problem
  - $\Rightarrow$ ensuring strict feasibility
  - $\Rightarrow$ Without affecting the optimality gap result
- This performance improvement was possible by considering the conservative approach
- Instead of the original problem, we actually solve a ν-tightened problem with a smaller constraint set.
- As long as the original problem is strongly feasible and we set ν appropriately
  - $\Rightarrow$ Such a tightening only leads to $\mathcal{O}(T^{-1/2})$ suboptimality
  - $\Rightarrow$ thus the overall optimality gap only changes by a constant factor.
- A regularization of the dual update is introduced in terms of problem constants
  - $\Rightarrow$ Similar tightest sub-optimality rate ($\mathcal{O}(T^{-1/2})$)
  - $\Rightarrow$ Ensuring null constraint violation in contrast to $\mathcal{O}(T^{-1/4})$ rate for existing settings

Pradhan, H., Bedi, A. S., Koppel, A., Rajawat, K. (2018, November). Exact nonparametric decentralized online optimization. In 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP) (pp. 643-647). IEEE.

Outline

- **Approach**:
  - ⇒ Hypothesized non-linear function in kernel Hilbert space
  - ⇒ Consider the conservative version (strict feasibility)
  - ⇒ Form stochastic lagrangian
  - ⇒ Apply stochastic primal dual method
  - ⇒ Take subspace projection (to handle memory growth)
- Sublinear convergence
  - ⇒ $\mathcal{O}(T^{-1/2})$ for primal optimality
  - ⇒ Zero constraint violation
- Generalizes existing rate results for primal-dual method
  - ⇒ to case of non-linear statistical models
- **Application**: Estimation of climatological fields
  - ⇒ Salinity and temperature measurement in Gulf of Mexico

The Problem

- Symmetric, connected and directed network of agents $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- Learning nonlinear statistical models is equivalent to finding
  $\Rightarrow f : \mathcal{X} \to \mathcal{Y}$, such that $y = f(\mathbf{x})$
- Loss $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ penalize deviations between $f(\mathbf{x})$, $\mathbf{y}$
- Encoded by a convex proximity function $h_{ij}(f_i(\mathbf{x}_i), f_j(\mathbf{x}_i))$
  $\Rightarrow$ incentivizes nearby agents to make similar decisions
- Yields the constrained functional stochastic program:

$$\mathbf{f}^\star = \underset{\{f_i\} \in \mathcal{H}}{\operatorname{argmin}} \; S(\mathbf{f}) := \sum_{i \in \mathcal{V}} \left(\mathbb{E}_{\mathbf{x}_i, y_i} \Big[\ell_i(f_i(\mathbf{x}_i), y_i) \Big] + \frac{\lambda}{2} \|f_i\|_{\mathcal{H}}^2 \right)$$

$$\text{s.t.} \quad H_{ij}(f_i, f_j) := \mathbb{E}_{\mathbf{x}_i} \Big[h_{ij}(f_i(\mathbf{x}_i), f_j(\mathbf{x}_i)) \Big] \leq \gamma_{ij}, \ \text{for all} \ \ j \in n_i. \tag{1}$$

Conservative Version of (1)

- Conservative version: 'ν' is added to the constraint in (1):

$$\mathbf{f}_\nu^\star = \operatorname*{argmin}_{\{f_i\}\in\mathcal{H}} S(\mathbf{f})$$
$$\text{s.t. } H_{ij}(f_i, f_j) + \nu \leq \gamma_{ij}, \text{ for all } j \in n_i, \tag{2}$$

- This allows us to establish approximate algorithmic solutions to (2)
  $\Rightarrow$ while ensuring constraints in (1) may be exactly satisfied.
- Optimality gap: $\mathcal{O}(T^{-1/2})$ (constraints satisfied in the long run).
- Note: Optimality gap not compromised as opposed to $\mathcal{O}(T^{-1/4})$

Lemma

For $0 \leq \nu \leq \xi/2$, it holds that:

$$S(\mathbf{f}_\nu^*) - S(f^*) \leq \mathcal{O}(\nu) \tag{3}$$

Mahdavi, M., Jin, R., Yang, T. (2012). Trading regret for efficiency: online convex optimization with long term constraints. The Journal of Machine Learning Research, 13(1), 2503-2528.

Learning with Representer Theorem

Corollary

Consider the sample average approximation of (1), and its associated Lagrangian relaxation. The each ith component of the solution to the resulting saddle-point problem can be expressed as

$$f_i^* = \sum_{t=1}^{T} w_{i,t} \kappa(\mathbf{x}_{i,t}, .) \tag{4}$$

where $w_{i,t}$ are real-valued coefficients.

- Stochastic augmented Lagrangian function of (2) at time t

$$\hat{\mathcal{L}}_t(f, \boldsymbol{\mu}) := \sum_{i \in \mathcal{V}} \Bigg[\ell_i(f_i(\mathbf{x}_{i,t}), y_{i,t}) + \frac{\lambda}{2} \|f_i\|_{\mathcal{H}}^2 + \sum_{j \in n_i} \Bigg\{ \Big[\mu_{ij}(h_{ij}(f_i(\mathbf{x}_{i,t}), f_j(\mathbf{x}_{i,t})) + \nu - \gamma_{ij}) \Big] - \frac{\delta\eta}{2} \mu_{ij}^2 \Bigg\} \Bigg] \tag{5}$$

where $\boldsymbol{\mu}$ is a lagrange multiplier, with μ_{ij} defined for each $(i, j) \in \mathcal{E}$.

Functional Derivative

- Functional stochastic gradient of local loss in (5):

$$\ell_i'(f_i(\mathbf{x}_{i,t}), y_{i,t}) := \nabla_{f_i} \ell_i(f_i(\mathbf{x}_{i,t}), y_{i,t})(\cdot) = \frac{\partial \ell_i(f_i(\mathbf{x}_{i,t}), y_{i,t})}{\partial f_i(\mathbf{x}_{i,t})} \frac{\partial f_i(\mathbf{x}_{i,t})}{\partial f_i}(\cdot) \tag{6}$$

- Using the reproducing property of the kernel we obtain

$$\frac{\partial f_i(\mathbf{x}_{i,t})}{\partial f_i} = \frac{\partial \langle f_i, \kappa(\mathbf{x}_{i,t}, \cdot)\rangle_{\mathcal{H}}}{\partial f_i} = \kappa(\mathbf{x}_{i,t}, \cdot) \tag{7}$$

- Now the full gradient result can be written as

$$\nabla_{f_i} \hat{\mathcal{L}}_t(f_t, \mu_t) = \ell_i'(f_i(\mathbf{x}_{i,t}), y_{i,t})\kappa(\mathbf{x}_{i,t}, \cdot) + \lambda f_i + \sum_{j \in n_i} \mu_{ij} h_{ij}'(f_i(\mathbf{x}_{i,t}), f_j(\mathbf{x}_{i,t}))\kappa(\mathbf{x}_{i,t}, \cdot) \tag{8}$$

Algorithm

- **loop in parallel** for agent $i \in \mathcal{V}$
- Observe local training example realization $(\mathbf{x}_{i,t}, y_{i,t})$
- Send $\mathbf{x}_{i,t}$ to the neighboring nodes, $j \in n_i$ and receive $f_{j,t}(\mathbf{x}_{i,t})$
- Receive $\mathbf{x}_{j,t}$ from the neighbouring nodes, $j \in n_i$ and send $f_{i,t}(\mathbf{x}_{j,t})$
- Stochastic primal descent step on Lagrangian:

$$f_{i,t+1}=f_{i,t}(1-\eta\lambda)-\eta\left[\ell_i'(f_{i,t}(\mathbf{x}_{i,t}),y_{i,t})+\sum_{j\in n_i}\mu_{ij}h_{ij}'(f_{i,t}(\mathbf{x}_{i,t}),f_{j,t}(\mathbf{x}_{i,t}))\right]\kappa(\mathbf{x}_{i,t},\cdot) \tag{9}$$

- Stochastic dual ascent step on Lagrangian:

$$\mu_{ij,t+1}=\left[\mu_{ij,t}(1-\delta\eta^2)+\eta\left(h_{ij}(f_{i,t}(\mathbf{x}_{i,t}),f_{j,t}(\mathbf{x}_{i,t}))-\gamma_{ij}+\nu\right)\right]_+ \tag{10}$$

Parametric update of weights and dictionary

- Using $f_{i,t}(\mathbf{x}) = \sum_{n=1}^{t-1} w_{i,n}\kappa(\mathbf{x}_{i,n}, \mathbf{x}) = \mathbf{w}_{i,t}^T \boldsymbol{\kappa}_{\mathbf{X}_{i,t}}(\mathbf{x})$, V parallel parametric updates on both kernel dictionaries $\mathbf{X}_i$ and $\mathbf{w}_i$ are

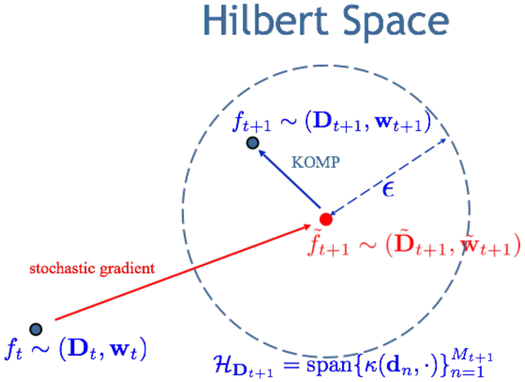
$$\mathbf{X}_{i,t+1} = [\mathbf{X}_{i,t}, \quad \mathbf{x}_{i,t}] \,, \tag{11}$$

$$[\mathbf{w}_{i,t+1}]_u = \begin{cases} (1-\eta\lambda)[\mathbf{w}_{i,t}]_u, \quad 0 \le u \le t-1 \\ -\eta\Big(\ell_i'(f_{i,t}(\mathbf{x}_{i,t}), y_{i,t}) \\ + \sum_{j\in n_i} \mu_{ij} h_{ij}'(f_{i,t}(\mathbf{x}_{i,t}), f_{j,t}(\mathbf{x}_{i,t}))\Big), u = t \end{cases}$$

- Data points $M_{i,t}$ grows by one each time (curse of kernelization).
- **Proj. Funct. Update**: Onto $\mathcal{H}_{\mathbf{D}_{i,t+1}} = \text{span}\{\kappa(\mathbf{d}_{i,n}, \cdot)\}_{n=1}^{M_{t+1}} \subset \mathcal{H}$

$$f_{i,t+1} := \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{i,t+1}}}\Big[f_{i,t} - \eta \nabla_{f_i} \hat{\mathcal{L}}_t(f_t, \mu_t)\Big]. \tag{12}$$

- Fix approximation error ϵ
- $\tilde{f}_{t+1} = f_t - \eta \nabla_{f_i} \hat{\mathcal{L}}_t(f_t, \mu_t)$
- Remove kernel element smallest error
- Project $\tilde{f}_{t+1}$ onto resulting RKHS
- Repeat until error is larger than ε



Non-asymptotic bound on Model Order

Theorem

Let $M_{i,t}$ denote the model order representing the number of dictionary elements in $\mathbf{D}_{i,t}$. Then with constant step size $\eta = 1/\sqrt{T}$ and compression budget ϵ, for a Lipschitz Mercer kernel κ on a compact set $\mathcal{X} \subset \mathbb{R}^p$, there exists a constant β such that for any training set $\{\mathbf{x}_{i,t}\}_{t=1}^{\infty}$, $M_{i,t}$ satisfies

$$M_{i,t} \leq \beta \left(\frac{R_M}{\alpha} \right)^{2p}, \tag{13}$$

where $\alpha = \epsilon/\eta$, $R_M = C + L_h E R_{i,t}$ and $R_{i,t} = \max_{j \in n_i} |\mu_{ij,t}|$. The total model order, M_t of the network consisting of N nodes is then

$$M_t = \sum_{i=1}^{N} M_{i,t}. \tag{14}$$

Convergence Analysis

Theorem

With $S(\mathbf{f})$ as the objective and $\mathbf{f}^*$ defined in (1), considering constant step-size $\eta = T^{-1/2}$, and $\nu = \zeta T^{-1/2} + \Lambda\alpha$, where $\zeta \geq \frac{1}{2}\left[R_{\mathcal{B}}^2 + (1+\delta)\left(2 + 2\left(\frac{4VR_{\mathcal{B}}(CX+\lambda R_{\mathcal{B}})}{\xi}\right)^2\right) + K\right]$ and $\Lambda \geq 4VR_{\mathcal{B}}$.

- *The average expected sub-optimality decays as*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[S(\mathbf{f}_t) - S(\mathbf{f}^\star)] \leq \mathcal{O}(T^{-1/2} + \alpha). \tag{15}$$

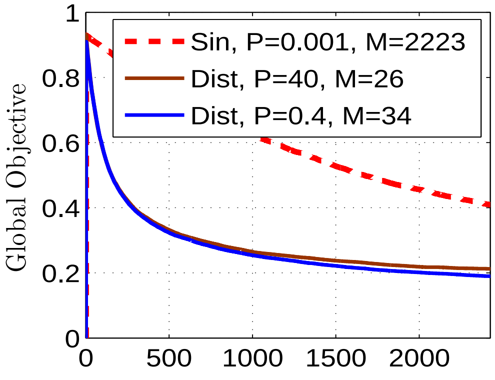
- *Moreover, the average of aggregate constraint is met, i.e.,*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[H_{ij}(f_{i,t}, f_{j,t}) - \gamma_{ij}\Big] \leq 0, \;\; \textit{for all} \;\; (i,j) \in \mathcal{E}. \tag{16}$$

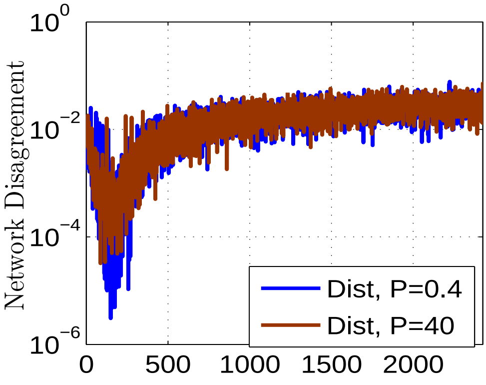
Estimating Temperature and Salinity of Ocean

- Climatological fields are obtained for a particular latitude and longitude in the Gulf of Mexico
  - $\Rightarrow$ for standard depths starting from 0 meters to 5000 meters
- The experiment is carried out considering 50 nodes
- Neighbouring node: if the distance is less than 1000 kilometers
- Proximity parameter: $\gamma_{ij} = \exp(-\text{dist}(i, j)/1000)$
- Step-size, $\eta = 0.01$ and regularizers λ, δ are set to 10^{-5}
- Bandwidth parameter of the Gaussian kernel is set at $\sigma = 50$
- Parsimony constant is fixed at two values, $P = 0.4$ and 40.
  - $\Rightarrow$ For centralized approach: $P = 0.001$

- Convergence of global objective and network disagreement



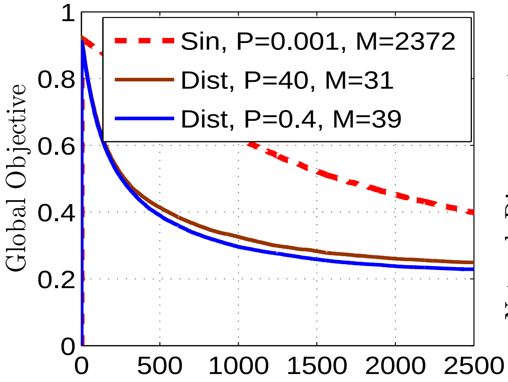
(a) Temperature (Global objective)



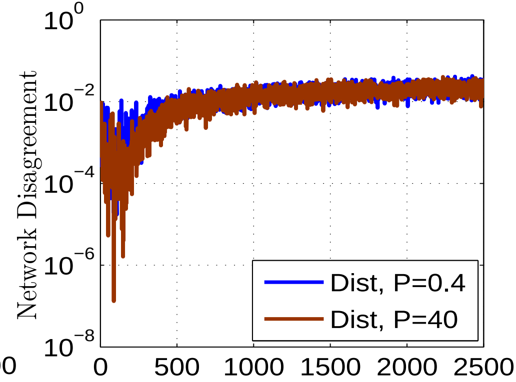
(b) Temperature(Network disag.)

Salinity Estimation

- Convergence of global objective and network disagreement



(c) Salinity (Global objective)



(d) Salinity (Network disag.)

Conclusion and Future Work

- Focused on online learning
  - ⇒ Decentralized heterogeneous networks
  - ⇒ Non-linear statistical models
  - ⇒ Conservative approach
  - ⇒ Optimality in terms of model complexity and iterations
- Proposed new variant of projected stochastic primal dual method
  - ⇒ Convergence to the optimum
  - ⇒ Finite growth of model order
  - ⇒ Observed good empirical performance
- **Future Work**:
  - ⇒ Asynchrony
  - ⇒ Reduce complexity of projections

Appendix: Supervised Learning Background

- $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ is random pair $\Rightarrow$ training examples
- $\ell : \mathcal{W} \to \mathbb{R}$ convex loss ($\mathcal{W} \subset \mathbb{R}^p$), merit of statistical model
- Find parameters $\mathbf{w}^* \in \mathbb{R}^p$ that minimize expected risk $L(\mathbf{w})$

$$\mathbf{w}^* := \operatorname*{argmin}_{\mathbf{w}} L(\mathbf{w}) := \operatorname*{argmin}_{\mathbf{w}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\mathbf{w}^\top \mathbf{x}, \mathbf{y})]$$

- Convex Optimization Problem for *linear statistical models*

  $\Rightarrow$ e.g., $y = \mathbf{w}^T \mathbf{x} \in \mathbb{R}$ or $y = \text{sgn}(\mathbf{w}^T \mathbf{x}) \in \{-1, 1\}$
- Solve with favorite descent method $\Rightarrow$ Good Performance

Appendix: Easy to Implement over Networks

- Symmetric, connected and directed network of agents $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- The nodes aims to make inferences from local data
- $|\mathcal{V}| = V$ nodes, $|\mathcal{E}| = M$ edges, and $n_i := \{j : (i,j) \in \mathcal{E}\}$
- Agent $i \in \mathcal{V}$ has a local copy of the classifier $\mathbf{w}_i$

  $\Rightarrow$ Observes some training examples $\Rightarrow (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X}_i \times \mathcal{Y}_i$

$$\mathbf{w}^* := \underset{\mathbf{w} \in \mathbb{R}^{p|\mathcal{V}|}}{\operatorname{argmin}} \sum_{i=1}^{|\mathcal{V}|} \mathbb{E}_{\mathbf{x}_i, y_i} \left[\ell(\mathbf{w}_i^\top \mathbf{x}_i, \mathbf{y}_i) \right]$$

$$s.t. \quad \mathbf{w}_i = \mathbf{w}_j \quad \text{for all} \quad j \in n_i$$

- Convex Optimization Problem for *linear statistical models*
- Solve with saddle point algorithms or penalty methods

  $\Rightarrow$ Can be implemented in a distributed fashion

Appendix: How to control the Model order?

- Project (9) onto a lower dimensional subspace $\mathcal{H}_{\mathbf{D}} \subseteq \mathcal{H}$
- $\mathcal{H}_{\mathbf{D}}$ is represented by a dictionary $\mathbf{D} = [\mathbf{d}_1, \ \dots, \ \mathbf{d}_M] \in \mathbb{R}^{p \times M}$.
- $\mathcal{H}_{\mathbf{D}} = \{f \ : \ f(\cdot) = \sum_{n=1}^{M} w_n \kappa(\mathbf{d}_n, \cdot) = \mathbf{w}^T \boldsymbol{\kappa}_{\mathbf{D}}(\cdot)\}$, $\{\mathbf{d}_n\} \subset \{\mathbf{x}_u\}_{u \le t}$.
- We denote the un-projected functional update as

$$\tilde{f}_{i,t+1} = f_{i,t} - \eta \nabla_{f_i} \hat{\mathcal{L}}_t(f_t, \mu_t) \ . \tag{17}$$

where $\nabla_{f_i} \hat{\mathcal{L}}_t(f_t, \mu_t) := \lambda f_{i,t} + \Big[\ell_i'(f_{i,t}(\mathbf{x}_{i,t}), y_{i,t}) + \sum_{j \in n_i} \mu_{ij} h_{ij}'(f_{i,t}(\mathbf{x}_{i,t}), f_{j,t}(\mathbf{x}_{i,t}))\Big] \kappa(\mathbf{x}_{i,t}, \cdot)$.

- $\tilde{f}_{i,t+1}$ in form of dictionary and coefficient vector:

$$\tilde{\mathbf{D}}_{i,t+1} = [\mathbf{D}_{i,t}, \ \ \mathbf{x}_{i,t}] \ ,$$

$$[\tilde{\mathbf{w}}_{i,t+1}]_u = \begin{cases} (1 - \eta\lambda)[\mathbf{w}_{i,t}]_u, & \text{for } 0 \le u \le t-1 \\ -\eta\Big(\ell_i'(f_{i,t}(\mathbf{x}_{i,t}), y_{i,t}) + \sum_{j \in n_i} \mu_{ij} h_{ij}'(f_{i,t}(\mathbf{x}_{i,t}), f_{j,t}(\mathbf{x}_{i,t}))\Big), & \text{for } u = t \end{cases}$$