Distributed Online Convex Optimization with Compressed Communication

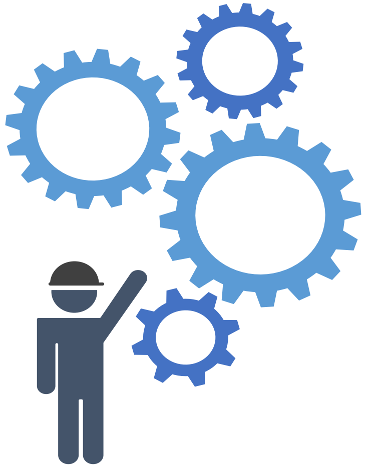
Presentation for NeurIPS 2022

Zhipeng Tu[1,2], Xi Wang[1,2], Yiguang Hong[*3], Lei Wang[4], Deming Yuan[5], Guodong Shi[2]

[1]AMSS, Chinese Academy of Sciences
[2]The University of Sydney
[3]Tongji University
[4]Zhejiang University
[5]Nanjing University of Science and Technology

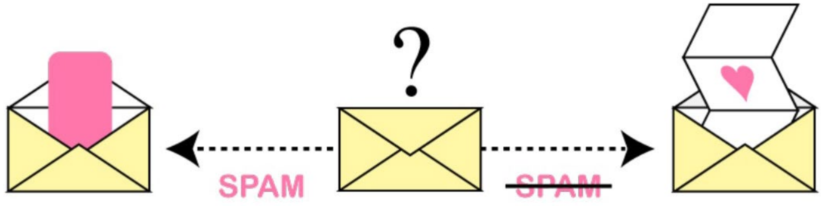
Presentation Outline

- Background
  - Distributed Optimization
  - Compressed Communication
  - Related Work
- Algorithms and Results
  - Full Information Feedback
  - One-point Bandit Feedback
  - Two-point Bandit Feedback
- Numerical Experiments
- Conclusions



Distributed Online Optimization

- Online tasks: streaming data are revealed incrementally, and decisions must be made before all data are available.
  - Spam filtering [Sculley and Wachman, SIGIR2007]
  - Dictionary learning [Mairal et al, ICML2009]
  - Advertising selection [Hazan et al, 2016]

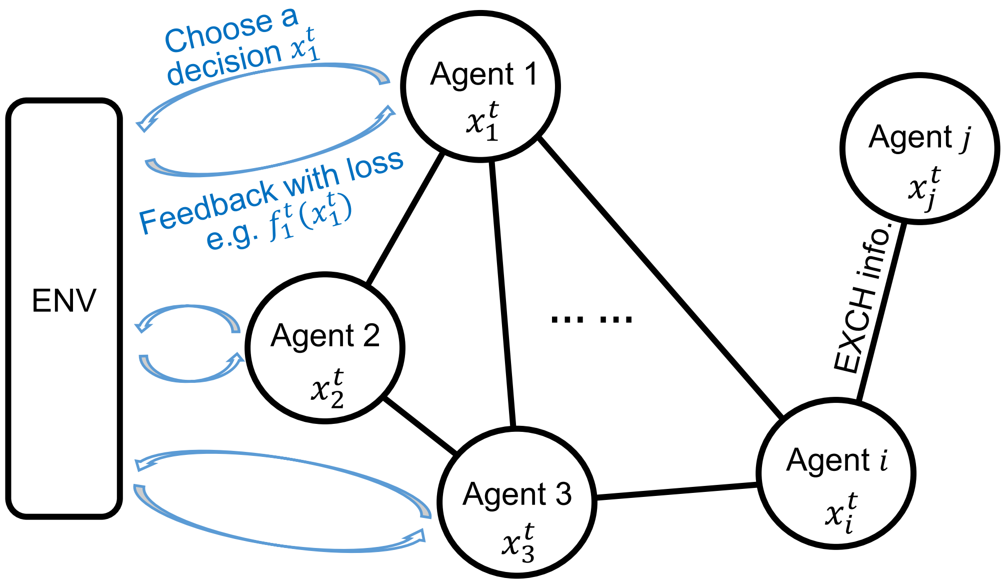


- Distributed setting: data collection, storage, and processing are performed in a multi-agent network.

- Goal: $\min_{x\in\Omega} \sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(x)$

  Metric: $\text{Regret}(j,T) = \sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(x_j^t) - \text{argmin}_x \sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(x)$

  No-regret: $\frac{\text{Regret}(T)}{T} \to 0$, as $T \to 0$



Compressed Communication

- Motivation: communication is a bottleneck!
  - High-dimensional data, large-scale network, limited bandwidth.
  - Data transmission is more time-consuming than calculation.
- Compressor: $Q(\cdot): \mathbb{R}^d \to \mathbb{R}^d$ is a mapping/operator whose output can be usually encoded with fewer bits.
- ω-contracted compressor: satisfying $\mathbb{E}_Q\|Q(x) - x\|^2 \le (1-\omega)\|x\|^2, \ \forall x \in \mathbb{R}^d$.

Example	description	ω	Bits to encode $Q(x)$
Sparsification [Stich et al, NeuIPS2018]	$\mathrm{Rand}_k, \quad \mathrm{Top}_k$	$\frac{k}{d}$	$kb + \log_2 d$
Random gossip [Koloskova et al, ICML2019]	$Q(x) = \begin{cases} x, & p \in [0,1] \\ 0, & \text{otherwise.} \end{cases}$	p	bdp
Random quantization [Alistarh et al, NeuIPS2017]	$\mathrm{QSGD}_s(x) = \frac{\mathrm{sgn}(x) \cdot \|x\|}{s\sigma} \circ \left\lfloor \frac{s\|x\|}{\|x\|} + \xi \right\rfloor$	$\frac{1}{\sigma}$	$\lceil \log_2(2s+1) \rceil d + b$

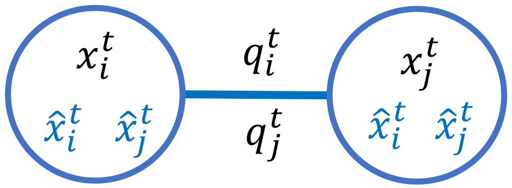
- Open problem:

whether it is possible to design provably no-regret distributed online algorithms that work with compressors.

Related Work

- Direct compression scheme: allows agents to compress their states and spread them directly.
  - fail to converge [Carli et al, ECC2007], [Aysal et al, TSP2008]
- Extrapolation compression scheme: allows agents to compress the extrapolation between the last two local states.
  - D-PSGD → ECD-PSGD [Tang et al, NeurIPS2018]
  - AMSGrad → ECD-AMSGrad [Li et al, CL2021] (online, empirical results)
- Difference compression scheme: allows agents to add replicas of neighboring states and compress the state-difference
  - D-PSGD → DCD-PSGD [Tang et al, NeurIPS2018]
  - SGD → CHOCO-SGD [Koloskova et al, ICML2019]
  - Event-trigger → SPARQ-SGD [Singh et al, TAC2022]
  - Gradient-tracking → C-GT [Liao et al, arXiv2021]
  - NIDS → COLD [Zhang et al, arXiv 2021]
  - EF → EF21 [Richtarik et al, NeurIPS2021]
  - Periodic averaging → FedPAQ [Reisizadeh et al, PMLR2020]

Difference-compressed communication



- $\hat{x}_i^t$ acts as a replica of x_i^t
- Compress the difference $q_i^t = Q(x_i^t - \hat{x}_i^t)$ and spread it
- Update $\hat{x}_i^{t+1} = \hat{x}_i^t + q_i^t$

✓ $\hat{x}_i^{t+1}$ actually tracks x_i^t

✓ difference → 0, compression error → 0

Full Information Feedback

The loss function f_i^t is revealed to node i at time t after the decision x_i^t is made.

We propose the DC-DOGD, which is based on DAOL [Yan et al, TKDE2012] and memory-efficient CHOCO-SGD [Koloskova et al, ICML2019].

Algo.1 Distributed Online Gradient Descent with Difference Compression (DC-DOGD)

Input: consensus stepsize γ, gradient descent stepsize $\{\eta_t\}_{t=1}^T$

Initialize: $x_i^1 = 0, \hat{x}_i^1 = 0, s_i^1 = 0, \forall i$

For $t = 1$ to T, **do** in parallel for each node i

Compress the difference $q_i^t = Q(x_i^t - \hat{x}_i^t)$, and update the local replica $\hat{x}_i^{t+1} = \hat{x}_i^t + q_i^t$.

Send q_i^t and receive q_j^t, and update the estimate of the consensus decision $s_i^{t+1} = s_i^t + \sum_{j=1}^N a_{ij}\, q_j^t$.

Calculate the gradient $g_i^t = \nabla f_i^t(x_i^t)$.

Update its decision variable $x_i^{t+1} = P_{\mathcal{K}}\left(x_i^t + \gamma\left(s_i^{t+1} - \hat{x}_i^{t+1}\right) - \eta_t g_i^t\right)$.

Difference compression

Observe the full function

Projection: remain in the feasible set

γ-gossip: renovate x_i^t towards the consensus decision

Gradient descent: minimize the local loss function

When there is no compression, DC-DOGD reduces to DAOL.

$$\hat{x}_i^{t+1} \to x_i^t, \quad s_i^{t+1} \to \sum_{j=1}^N a_{ij}\, x_j^t, \quad x_i^{t+1} \xrightarrow{\gamma=1} P_{\mathcal{K}}\left(\sum_{j=1}^N a_{ij}\, x_j^t - \eta_t \nabla f_i^t(x_i^t)\right)$$

Full Information Feedback

Assumptions

- 1. The connectivity matrix A is symmetric doubly stochastic.
- 2. Q is ω-contracted.
- 3. The convex set $\mathcal{K}$ is bounded with diameter D.
- 4. f_i^t is convex and differentiable with bounded gradient. $\max_{i,t,x} \|\nabla f_i^t(x)\| \leq G$.
- 5. f_i^t is μ-strongly convex.

Full Information Feedback

Theorem 1 (DC-DOGD)

Take $\gamma = \frac{3\delta^3\omega^2(\omega+1)}{48(\delta^2+18\delta\beta^3+36\beta^2)\beta^2(\omega+2)(1-\omega)+4\delta^2(\beta^2+\beta)\left((\omega+2)(1-\omega)\right)\omega+6\delta^3\omega}$, where $\delta := 1 - |\lambda_2(A)|, \beta := \|I - A\|_2$.

(i) (Convex case) Under Assumptions 1,2,3,4. Take $\eta_t = \frac{D}{G\sqrt{t+c}}$, for a constant $c \geq \frac{8}{3\gamma\delta}$, then

$$\mathbb{E}_Q[\mathrm{Regret}(j,T)] \leq \left(\frac{1}{2} + 8\sqrt{3}\left(\sqrt{N} + \frac{2\sqrt{3}}{\gamma\delta} + 1\right)\left(1 + \frac{1}{\gamma\delta} + \frac{1}{\omega}\right)\right)NGD\sqrt{T+c} = \mathcal{O}\left((\omega^{-2}N^{1/2} + \omega^{-4})N\sqrt{T}\right).$$

(ii) (Strongly convex case) Under Assumptions 1,2,4,5. Take $\eta_t = \frac{1}{\mu(t+c)}$, for a constant $c \geq \frac{16}{3\gamma\delta}$, then

$$\mathbb{E}_Q[\mathrm{Regret}(j,T)] \leq 4\sqrt{3}\left(\sqrt{N} + \frac{2\sqrt{3}}{\gamma\delta} + 1\right)\left(1 + \frac{1}{\gamma\delta} + \frac{1}{\omega}\right)\frac{NG^2}{\mu}\ln(T+c) = \mathcal{O}\left((\omega^{-2}N^{1/2} + \omega^{-4})N\ln T\right).$$

One-point Bandit Feedback

After making the decision x_i^t at time t, agent i can only query the loss function value at one point around x_i^t.

We propose the DC-DOBD, which follows DC-DOGD.

Algo.2 Distributed Online One-point Bandit Gradient Descent with Difference Compression (DC-DOBD)

Input: γ, $\{\eta_t\}_{t=1}^{T}$, exploration parameter ϵ, shrinkage parameter ς

Initialize: $x_i^1 = 0, \hat{x}_i^1 = 0, s_i^1 = 0, \forall i$

For $t = 1$ to T, **do** in parallel for each node i

Compress the difference $q_i^t = Q(x_i^t - \hat{x}_i^t)$, and update the local replica $\hat{x}_i^{t+1} = \hat{x}_i^t + q_i^t$.

Send q_i^t and receive q_j^t, and update the estimate of the consensus decision $s_i^{t+1} = s_i^t + \sum_{j=1}^{N} a_{ij}\, q_j^t$.

Choose a unit-norm vector $u_i^t \in \mathbb{R}^d$ at random, and construct the gradient estimator $g_i^t = \frac{d}{\epsilon} f_i^t(x_i^t + \epsilon u_i^t) u_i^t$.

Update its decision variable $x_i^{t+1} = P_{(1-\varsigma)\mathcal{K}}\left(x_i^t + \gamma\left(s_i^{t+1} - \hat{x}_i^{t+1}\right) - \eta_t g_i^t\right)$.

$\mathbb{E}_u[g_i^t] = \nabla \hat{f}_i^t(x)$

[Flaxman et al, SIAM2005]

DC-DOBD actually performs the gradient descent on the function $\hat{f}_i^t(x) = \mathbb{E}_u[x + \epsilon u]$ restricted to the convex set $(1-\varsigma)\mathcal{K}$.

One-point Bandit Feedback

Assumptions

3. $\mathcal{K}$ is bounded with diameter D.

4. f_i^t is differentiable with bounded gradient.

Assumptions

6. $r\mathcal{B} \subseteq \mathcal{K} \subseteq R\mathcal{B}, \mathcal{B} = \{u \in \mathbb{R}^d : \|u\| \leq 1\}$.

7. f_i^t is l-Lipschitz continuous.

$\max_{i,t,x} |f_i^t(x)| \leq B$

Theorem 2 (DC-DOBD)

Denote $H = 4\sqrt{3}\left(\sqrt{N} + \frac{2\sqrt{3}}{\gamma\delta} + 1\right)\left(1 + \frac{1}{\gamma\delta} + \frac{1}{\omega}\right)$. γ is chosen as in Theorem 1.

(i) (Convex case) Under Assumptions 1,2,6,7. Take $\eta_t = \frac{2R\epsilon}{dB\sqrt{t+c}}$, for $c \geq \frac{8}{3\gamma\delta}$, and $\epsilon = \left(\frac{(1+4H)dBR}{2(l+B/r)}\right)^{\frac{1}{2}} \frac{(T+c)^{\frac{1}{2}}}{T^{\frac{1}{2}}}$, $\varsigma = \frac{\epsilon}{r}$, then

$$\mathbb{E}[\text{Regret}(j,T)] \leq 2NT^{\frac{1}{2}}(T+c)^{\frac{1}{4}}\sqrt{2(1+4H)\left(l + \frac{B}{r}\right)dBR} = \mathcal{O}\left(d^{\frac{1}{2}}N^{\frac{5}{4}}T^{\frac{3}{4}}\right).$$

(ii) (Strongly convex case) + Assumption 5. Take $\eta_t = \frac{1}{\mu(t+c)}$, for $c \geq \frac{16}{3\gamma\delta}$, and $\epsilon = \left(\frac{Hd^2B^2\ln(T+c)}{(l+B/r)\mu T}\right)^{\frac{1}{3}}$, $\varsigma = \frac{\epsilon}{r}$, then

$$\mathbb{E}[\text{Regret}(j,T)] \leq 3N\left(\frac{Hd^2B^2}{\mu}\right)^{\frac{1}{3}}\left(l + \frac{B}{r}\right)^{\frac{2}{3}} T^{\frac{2}{3}}\ln^{\frac{1}{3}}(T+c) = \mathcal{O}\left(d^{\frac{2}{3}}N^{\frac{7}{6}}T^{\frac{2}{3}}\ln^{\frac{1}{3}}T\right).$$

Two-point Bandit Feedback

After making the decision x_i^t at time t, agent i can query the loss function value at two points around x_i^t.

We propose the DC-DO2BD as a variant of DC-DOBD.

Algo.3 Distributed Online Two-point Bandit Gradient Descent with Difference Compression (DC-DO2BD)

Input: γ, $\{\eta_t\}_{t=1}^T$, exploration parameter ϵ, shrinkage parameter ς

Initialize: $x_i^1 = 0, \hat{x}_i^1 = 0, s_i^1 = 0, \forall i$

For $t = 1$ to T, **do** in parallel for each node i

Compress the difference $q_i^t = Q(x_i^t - \hat{x}_i^t)$, and update the local replica $\hat{x}_i^{t+1} = \hat{x}_i^t + q_i^t$.

Send q_i^t and receive q_j^t, and update the estimate of the consensus decision $s_i^{t+1} = s_i^t + \sum_{j=1}^N a_{ij}\, q_j^t$.

Choose a unit-norm vector $u_i^t \in \mathbb{R}^d$ at random, and construct the gradient estimator $g_i^t = \frac{d}{2\epsilon}\left(f_i^t(x_i^t + \epsilon u_i^t) - f_i^t(x_i^t - \epsilon u_i^t)\right) u_i^t$.

Update its decision variable $x_i^{t+1} = P_{(1-\varsigma)\mathcal{K}}\left(x_i^t + \gamma\left(s_i^{t+1} - \hat{x}_i^{t+1}\right) - \eta_t g_i^t\right)$.

$\mathbb{E}_u[g_i^t] = \nabla \hat{f}_i^t(x)$

[Agarwal et al, COLT2010]

Two-point Bandit Feedback

$$\mathrm{Regret}_2(j,T)=\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{f_i^t\left(x_j^t+\epsilon u_j^t\right)-f_i^t\left(x_j^t-\epsilon u_j^t\right)}{2}-\sum_{t=1}^{T}\sum_{i=1}^{N}f_i^t(x^*)$$

Theorem 3 (DC-DO2BD)

γ and H are defined as before.

(i) (Convex case) Under Assumptions 1,2,6,7. Take $\eta_t=\frac{2R}{dl\sqrt{t+c}}$, for $c\geq\frac{8}{3\gamma\delta}$, and $\epsilon=\frac{1}{\sqrt{T}}$, $\varsigma=\frac{\epsilon}{r}$, then

$$\mathbb{E}[\mathrm{Regret}_2(j,T)]\leq(1+4H)RNdl\sqrt{T+c}+\left(3+\frac{2R}{r}\right)Ndl\sqrt{T}=\mathcal{O}\left(\left(\omega^{-2}N^{1/2}+\omega^{-4}\right)Nd\sqrt{T}\right).$$

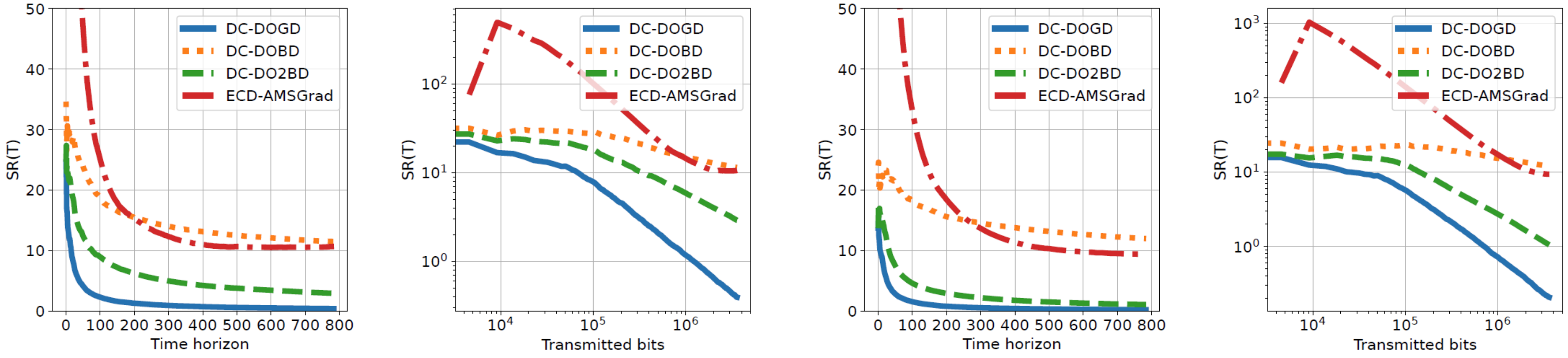
(ii) (Strongly convex case) + Assumption 5. Take $\eta_t=\frac{1}{\mu(t+c)}$, for $c\geq\frac{16}{3\gamma\delta}$, and $\epsilon=\frac{\ln T}{T}$, $\varsigma=\frac{\epsilon}{r}$, then

$$\mathbb{E}[\mathrm{Regret}_2(j,T)]\leq\frac{1}{\mu}Nd^2l^2H\ln(T+c)+\left(3+\frac{2R}{r}\right)Ndl\ln T=\mathcal{O}\left(\left(\omega^{-2}N^{1/2}+\omega^{-4}\right)Nd^2\ln T\right).$$

Numerical Experiments

- Task: diabetes prediction
- Dataset: *diabetes-binary-BRFSS2015* (70692 instances, 21 features, 2 labels)
- Model: distributed online regularized logistic regression with the local loss function:

$$f_i^t(x) = \sum_{j=1}^{S} \log\left(1 + \exp\left(-b_{i,j}^t \langle a_{i,j}^t, x\rangle\right)\right) + \frac{\mu}{2}\|x\|^2$$



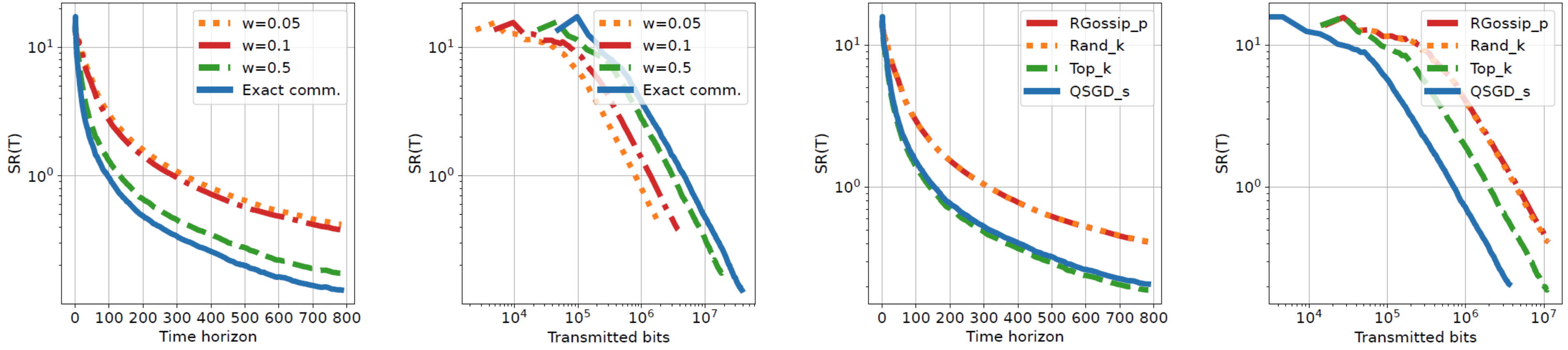
(a) Convex case (b) Strongly convex case

Figure 1: Comparison of algorithms DC-DOGD, DC-DOBD, DC-DO2BD, and ECD-AMSGrad with $\mathrm{QSGD}_2, \omega = 0.3, \mathcal{G}(9, 18)$.

✓ DC-DOGD, DC-DOBD, and DC-DO2BD are no-regret.

✓ DC-DOGD and DC-DO2BD significantly outperform ECD-AMSGrad.

Numerical Experiments

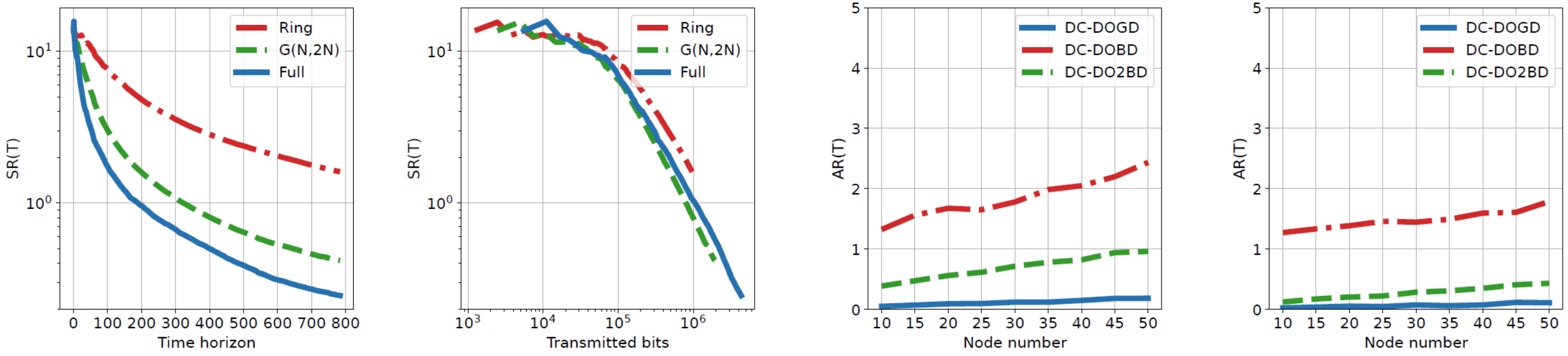


(a) Compression ratio (b) Compressor type

Figure 2: The impacts of compression ratio and compressor type for DC-DOGD over $\mathcal{G}(9, 18)$ in the strongly convex case.

- ✓ Effectively reduce the total transmitted bits for distributed online training.
- ✓ e.g. DC-DOGD with ω = 0.05 have approximately 8× reduction on transmitted bits to reach a certain average regret compared with DAOL.

Numerical Experiments



(a) Different topologies (b) Node number(left: convex, right: strongly convex)

Figure 3: The impacts of topology and node number.

Conclusions

- We propose communication-efficient distributed online algorithms for the cases of full information feedback (DC-DOGD), one-point bandit feedback (DC-DOBD), and two-point bandit feedback (DC-DO2BD), respectively.
- We make the technical advance to combine the difference compression scheme with the projection scheme. Through proper design, the errors can be estimated and controlled by γ and η_t.
- We analyze the regret bounds of the proposed algorithms for convex and strongly convex losses. The obtained regret bounds match those of uncompressed algorithms w.r.t T. Our algorithms are no-regret with theoretical guarantees.
- We give exhaustive experiments. The proposed algorithms can reduce the total transmitted bits for distributed online training.

Table 1: Regret bounds in different settings

Settings	convex losses	strongly convex losses
Full information	$\mathcal{O}\left(\left(\omega^{-2}N^{1/2}+\omega^{-4}\right)N\sqrt{T}\right)$	$\mathcal{O}\left(\left(\omega^{-2}N^{1/2}+\omega^{-4}\right)N\ln(T)\right)$
One-point bandit	$\mathcal{O}\left(\left(\omega^{-2}N^{1/2}+\omega^{-4}\right)^{1/2}Nd^{1/2}T^{3/4}\right)$	$\mathcal{O}\left(\left(\omega^{-2}N^{1/2}+\omega^{-4}\right)^{1/3}Nd^{2/3}T^{2/3}\ln^{1/3}(T)\right)$
Two-point bandit	$\mathcal{O}\left(\left(\omega^{-2}N^{1/2}+\omega^{-4}\right)Nd\sqrt{T}\right)$	$\mathcal{O}\left(\left(\omega^{-2}N^{1/2}+\omega^{-4}\right)Nd^2\ln(T)\right)$

References

[Sculley and Wachman, SIGIR2007] David Sculley and Gabriel M Wachman. Relaxed online svms for spam filtering. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 415–422, 2007.

[Mairal et al, ICML2009] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 689–696, 2009.

[Hazan et al, 2016] Elad Hazan et al. Introduction to online convex optimization. Foundations and Trends® in Optimization, 2(3-4):157–325, 2016.

[Carli et al, ECC2007] Ruggero Carli, Fabio Fagnani, Paolo Frasca, Tom Taylor, and Sandro Zampieri. Average consensus on networks with transmission noise or quantization. In 2007 European Control Conference, pages 1852–1857. IEEE, 2007.

[Aysal et al, TSP2008] Tuncer Can Aysal, Mark J Coates, and Michael G Rabbat. Distributed average consensus with dithered quantization. IEEE Transactions on Signal Processing, 56(10):4905–4918, 2008.

[Tang et al, NeurIPS2018] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. Advances in Neural Information Processing Systems, 31, 2018.

[Li et al, CL2021] Guangxia Li, Jia Liu, Xiao Lu, Peilin Zhao, Yulong Shen, and Dusit Niyato. Decentralized online learning with compressed communication for near-sensor data analytics. IEEE Communications Letters, 25(9):2958–2962, 2021.

[Koloskova et al, ICML2019] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In International Conference on Machine Learning, pages 3478–3487. PMLR, 2019.

[Alistarh et al, NeuIPS2017] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communicationefficient sgd via gradient quantization and encoding. Advances in Neural Information Processing Systems, 30, 2017.

References

[Stich et al, NeuIPS2018] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. Advances in Neural Information Processing Systems, 31, 2018.

[Singh et al, TAC2022] Navjot Singh, Deepesh Data, Jemin George, and Suhas Diggavi. Sparq-sgd: Event-triggered and compressed communication in decentralized optimization. IEEE Transactions on Automatic Control, 2022.

[Liao et al, arXiv2021] Yiwei Liao, Zhuorui Li, Kun Huang, and Shi Pu. Compressed gradient tracking methods for decentralized optimization with linear convergence. arXiv preprint arXiv:2103.13748, 2021.

[Zhang et al, arXiv 2021] Jiaqi Zhang, Keyou You, and Lihua Xie. Innovation compression for communication-efficient distributed optimization with linear convergence. arXiv preprint arXiv:2105.06697, 2021.

[Richtarik et al, NeurIPS2021] Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. Advances in Neural Information Processing Systems, 34, 2021.

[Reisizadeh et al, PMLR2020] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq:A communication-efficient federated learning method with periodic averaging and quantization. In International Conference on Artificial Intelligence and Statistics, pages 2021–2031. PMLR, 2020.

[Yan et al, TKDE2012] Feng Yan, Shreyas Sundaram, SVN Vishwanathan, and Yuan Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. IEEE Transactions on Knowledge and Data Engineering, 25(11):2483–2493, 2012.

[Flaxman et al, SIAM2005] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 385–394, 2005.

[Agarwal et al, COLT2010] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In Proceedings of the 23rd Annual Conference on Learning Theory, pages 28–40. Citeseer, 2010.

Thank You!