On the Interplay between Statistics, Computation and Communication in Decentralised Learning

Patrick Rebeschini

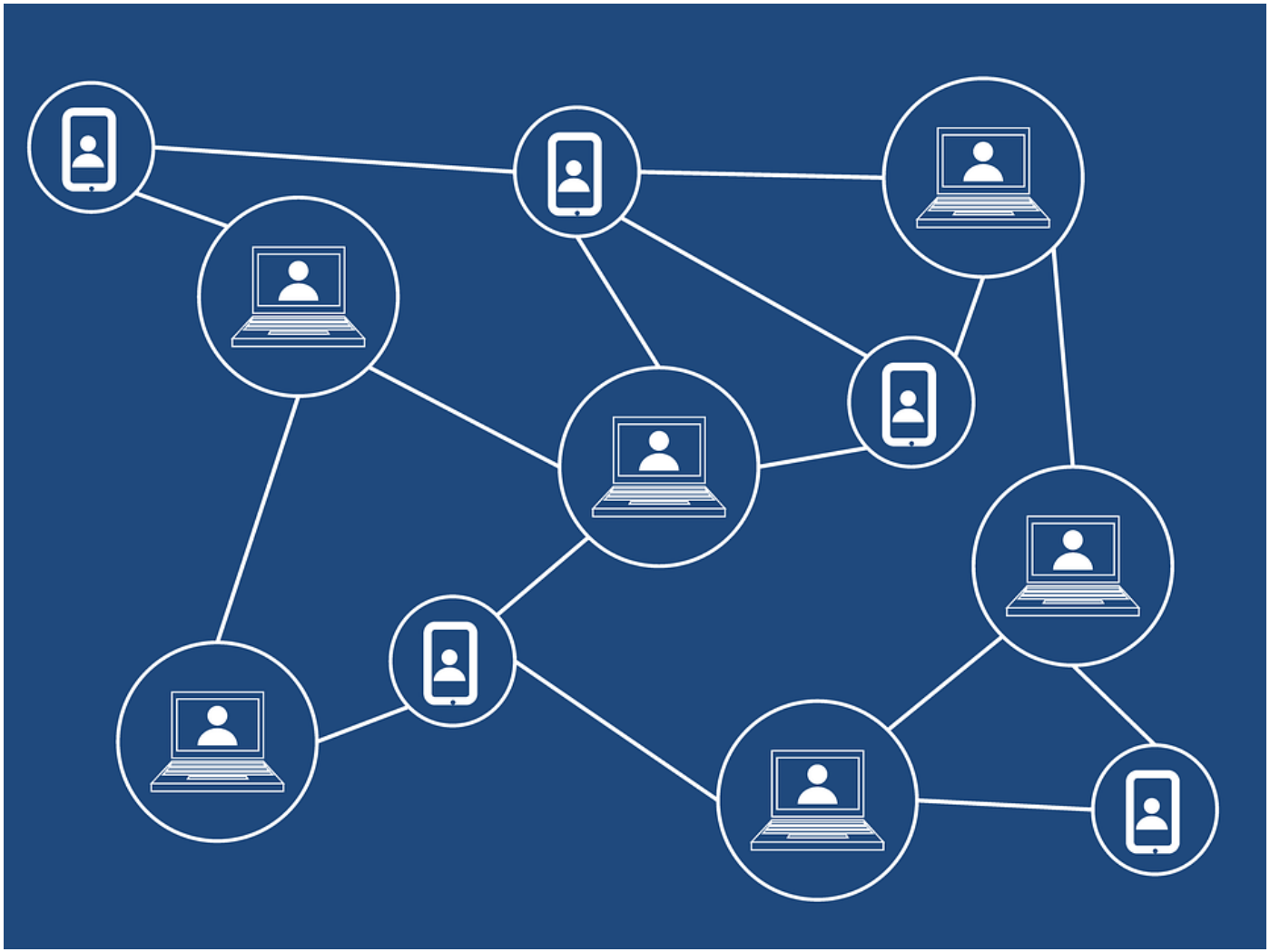
(joint works with D. Richards and with D. Martínez-Rubio and V. Kanade)

DEPARTMENT OF
STATISTICS

The Optimization and Statistical Learning workshop
Les Houches School of Physics

March 29th 2019

Decentralized Learning



- **Data is collected in different places:** sensor networks, control, online
- **Data is splitted:** computation/storage, bandwidth limitations, robustness

Setting

- Fixed graph with N nodes (agents)
- Agents given M data samples each; solve the **same statistical** problem
- Agents **collaborate** by iteratively sharing local models (not data) (synchronous, lockstep, edge communication delay τ)
- ***Desiderata:***
  - **Statistics:** recover optimal guarantees w.r.t. NM samples
  - **Runtime:** speed-up due to parallel computing, ideally by a factor N
  - **Communication:** fixed cost per agent per step, ideally independent of N

Q. Interplay statistics, computation, communication?
Is there a trade-off between these resources or not?

Offline Learning

(joint work D. Richards)

Single-Machine Learning

$Z_1, \ldots, Z_{MN} \sim \mathbf{P}$

($\mathbf{P}$ unknown)

Goal

Minimize expected risk: $r(\omega) = \mathbf{E}\,\ell(\omega, Z)$

Offline Learning

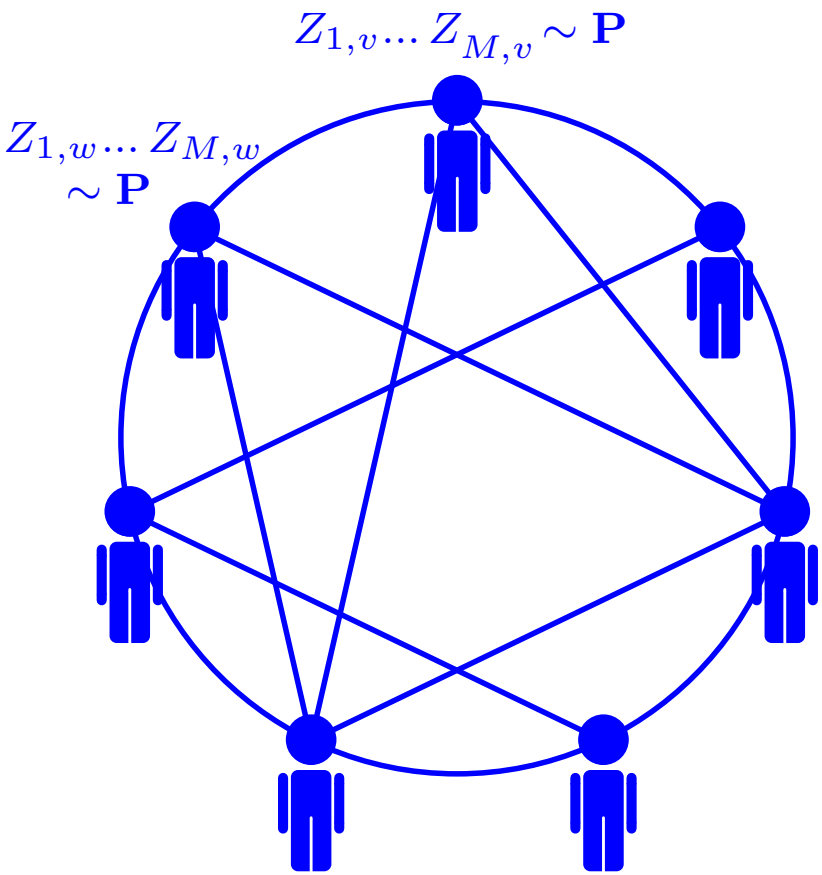
Data available at once

Empirical Risk: $R(\omega) = \frac{1}{MN} \sum_{i=1}^{MN} \ell(\omega, Z_i)$

Algorithm: $\omega_t \in \Omega$ to minimize Emp. Risk

Excess Risk: $\boxed{r(\omega_t) - \inf_{\omega} r(\omega)}$

Decentralized Learning



Goal

Minimize expected risk: $r(\omega) = \mathbf{E}\,\ell(\omega, Z)$

Offline Learning

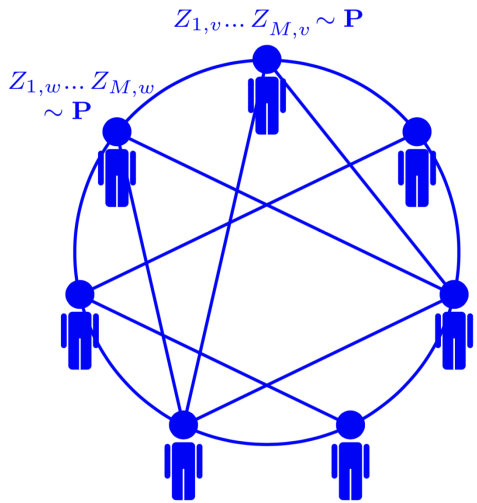
Data available at once

Emp. Risk: $R(\omega) = \frac{1}{N} \sum_{v \in V} \frac{1}{M} \sum_{i=1}^{M} \ell(\omega, Z_{i,v})$

Algorithm: $\omega_{t,v} \in \Omega$ to minimize Emp. Risk

Excess Risk: $\sup_{v \in V} r(\omega_{t,v}) - \inf_{\omega} r(\omega)$

Decentralized Optimization and Statistics



Communication Matrix $P \in \mathbb{R}^{N \times N}$:

- Supported on graph
- $P^t \xrightarrow[t \to \infty]{} \mathbf{1}\mathbf{1}^\top / N \iff$ (Boyd and Xiao, 2004)

$$P\mathbf{1} = \mathbf{1} \qquad \mathbf{1}^\top P = \mathbf{1}^\top \qquad \lambda_2 < 1$$

$$\Theta\left(\frac{1}{1-\lambda_2}\right) = \begin{cases} N^2 & \text{Cycle} \\ N & \text{Grid} \\ 1 & \text{Complete} \end{cases}$$

- Growing literature on **consensus optimization** for machine learning:

$$\text{minimize} \quad R(\omega) = \frac{1}{N} \sum_{v \in V} f_v(\omega)$$

$$\text{subject to} \quad \omega \in \Omega$$

- Statistically motivated by excess risk decomposition: (Bousquet and Bottou, 08)

$$\text{Excess Risk} \leq 2 \underbrace{\sup_{\omega \in \Omega} |r(\omega) - R(\omega)|}_{\text{Statistics}} + \underbrace{\sup_{v \in V} R(\omega_{t,v}) - \inf_{\omega \in \Omega} R(\omega)}_{\text{Optimization}} + \text{Approxim.}$$

Runtime of Decentralized/Consensus Optimization

Single-Step methods: (single consensus step between gradient computations)

- Distributed SGD (Nedic & Ozdaglar, 2009) $O\big((M+\tau+\deg(P))\frac{N^3R^2L^2}{\varepsilon^2}\big)$
- Decentralized Dual Averaging (Duchi et al., 2012) $O\big((M+\tau+\deg(P))\frac{R^2L^2}{(1-\lambda_2)\varepsilon^2}\big)$
- D-ADMM $O\big((M+\tau+\deg(P))\frac{2\kappa_\ell^2}{\sqrt{1+4\kappa_\ell^2(1-\lambda_2)}-1}\big)$

  (Boyd et al., 2011; Wei & Ozdaglar, 2012; Shi et al., 2014; Lutzeler et al., 2016)
- EXTRA algorithm $\exists\delta>0$ s.t. $O\big((M+\tau+\deg(P))\delta\log\frac{1}{\varepsilon}\big)$

  (Shi et al., 2015; Mokhtari & Ribeiro, 2016)
- Augmented Lagrangians (Jakovetić et al., 15) $O\big((M+\tau+\deg(P))\frac{2\kappa_\ell^2}{\sqrt{1+4\kappa_\ell^2(1-\lambda_2)}-1}\big)$
- DIGing (Nedich et al., 2016) $O\big((M+\tau+\deg(P))N^{9/2}\kappa_\ell^{3/2}\log\frac{1}{\varepsilon}\big)$
- ...

Multi-Step methods: (multiple consensus steps between gradient computations)

- Multi-Agent Mirror Descent (Rabbat, 2015) $O\big((M+\frac{\tau+\deg(P)}{1-\lambda_2})\frac{\sigma^2}{\mu_\Psi\varepsilon}\big)$
- Multi-Step Dual Accelerated (matching lower bounds)

  (Scaman et al., 2017) $O\big((M+\frac{\tau+\deg(P)}{\sqrt{1-\lambda_2}})\sqrt{\kappa_\ell}\log\frac{1}{\varepsilon}\big)$

  (Scaman et al., 2018) $O\big(\frac{\tau+\deg(P)}{\sqrt{1-\lambda_2}}\frac{RL}{\varepsilon}+M(\frac{RL}{\varepsilon})^2\big)$

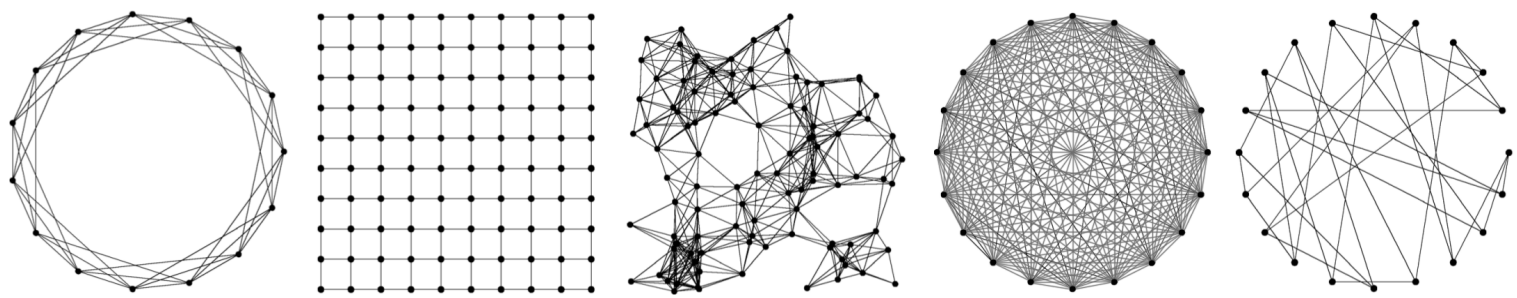
Single-Step Decentralized Optimization

Algorithms that alternate single computation and communication steps have dependence on $\frac{1}{1-\lambda_2}$ (or $\frac{1}{\sqrt{1-\lambda_2}}$ for gossip-accelerated methods). They achieve:

- Decentralized (Duchi et al., 2012): $O\big((M+\tau+\deg(P))\frac{R^2L^2}{(1-\lambda_2)\varepsilon^2}\big)$
- Single-Machine: $O\big(NM\frac{R^2L^2}{\varepsilon^2}\big)$

If $M \gtrsim \tau + \deg(P)$ then **Speedup** $= \frac{\text{runtime single-machine}}{\text{runtime decentralized}} = O((1-\lambda_2)N)$

	Cycle	Grid	Random Geometric†	Complete	Expander
Speedup	$O(1/N)$	$O(1)$	$O(1)$	$O(N)$	$O(N)$
Communication	$O(1)$	$O(1)$	$O(1)$	$O(N)$	$O(1)$



†connectivity radius $> \sqrt{15\log N/N} \Rightarrow$ connected and bounded degree

Multi-Step Decentralized Optimization

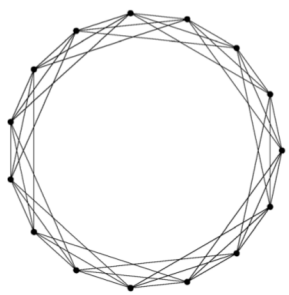
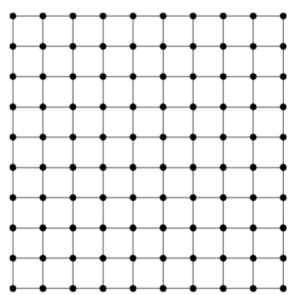
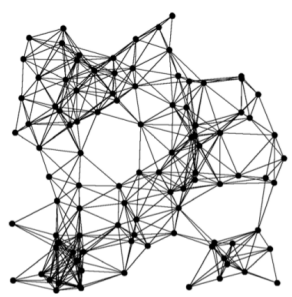
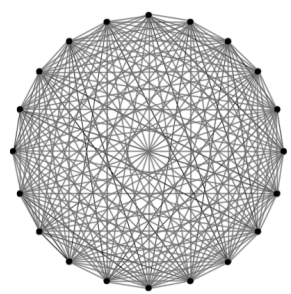
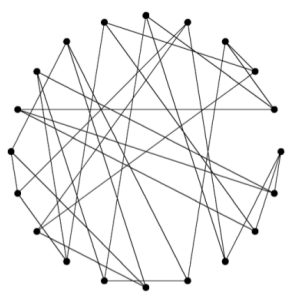
Extra communication (*multi-step* methods) yields optimal runtime algorithms:

- Decentralized (Scaman et al., 2017): $O((M + \frac{\tau+\deg(P)}{\sqrt{1-\lambda_2}})\sqrt{\kappa_\ell}\log\frac{1}{\varepsilon})$
- Single-Machine (Nesterov, 2004): $O(NM\sqrt{\kappa_g}\log\frac{1}{\varepsilon})$

If $M \gtrsim \frac{\tau+\deg(P)}{\sqrt{1-\lambda_2}}$ **(graph-dependent)**, $\kappa_\ell \approx \kappa_g$ $(\kappa_\ell \geq \kappa_g)$ then **Speedup** $= O(N)$

However: $\left\lfloor \frac{1}{\sqrt{1-\lambda_2}} \right\rfloor$ communication rounds per gradient computation step

	Cycle	Grid	Random Geometric	Complete	Expander
Speedup	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$
Communication	$O(N)$	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(N)$	$O(1)$

Decentralized Optimization: Trade-Off

Consensus Optimization:

Graphs with smaller spectral gap benefit less from decentralization

Trade-off speedup and communication

Can statistics help?

Statistics and Implicit Regularization

Consensus Problem

$$\text{minimize} \quad \frac{1}{N}\sum_{v\in V} f_v(\omega)$$

$$\text{subject to} \quad \omega \in \Omega$$

Implicit Regularization

__Tune__ algorithm to minimize

$$\frac{1}{N}\sum_{v\in V}\frac{1}{M}\sum_{i=1}^{M}\ell(\omega, Z_{v,i})$$

Limitations of consensus methods for decentralized learning:

- Designed for arbitrary local functions f_v (Shamir and Srebro, 2014), but

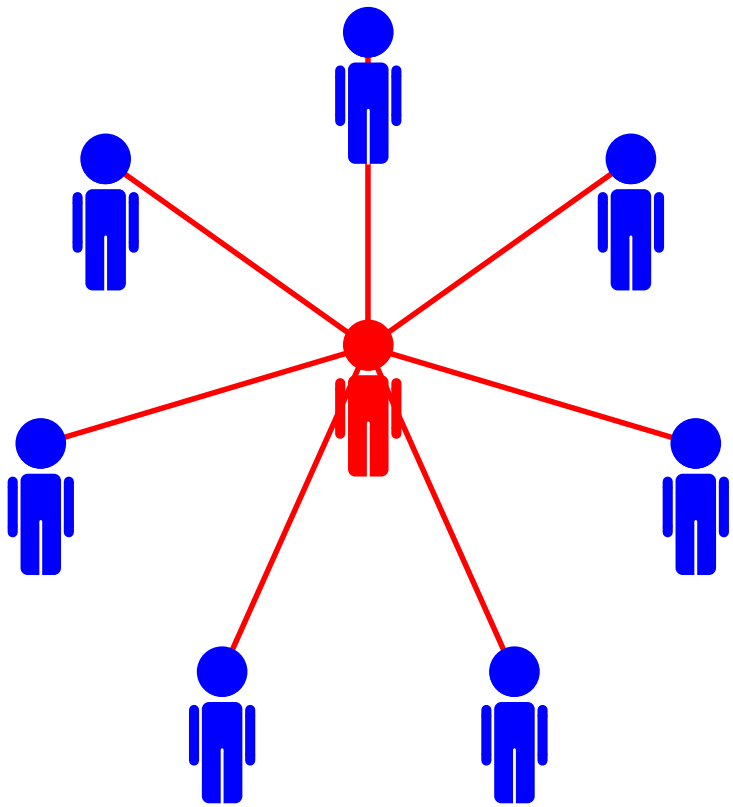
$$f_v(\omega) = \frac{1}{M}\sum_{i=1}^{M}\ell(\omega, Z_{i,v}) \qquad f_v \approx f_w \text{ when } M \text{ large}$$

IDEA: Exploit statistics (concentration)
INTUITION: more samples $\Rightarrow$ less communication

- Designed to minimize regularized training error
IDEA: Perform implicit regularization (no constraints/penalization)
Compute regularization path on the fly (time is regularizer). Tractability

Centralized Learning

Distributed statistical learning investigated primarily in **centralized** settings



- Most papers in decentralized setting only consider training error (no test error)
- Some works in the single-pass/online setting (do not exploit concentration): (Rabbat, 2015), (Bijral et al, 2017), (Lian et al., 2017), (Assran et al, 2018)

Centralized Learning

Centralized algorithms (Empirical Risk Minimization + Explicit Regularization):

- **One-shot averaging (parametric):** (Zinkevich et al., 10) (Zhang et al., 13)
  Only 1 round of communication. Does not reduce bias (Shamir et al., 14)
- **One-shot averaging (nonparametric):** (Zhang et al., 2013), (S. Lin and Guo et al., 2017), (J. Lin et al., 2018)
  Only 1 round of communication. Saturation: optimal stat. bounds $r \leq 1$
- **Iterative (second-order):** DANE (Shamir et al., 2014), DiSCO (Zhang et al., 2015), AIDE (Reddi et al., 2016)
  Communication efficient (exploit concentration of local objectives)
  No linear speedup (cost of inverting matrices does not parallelize)
- **Iterative (first-order):** DSVRG (Lee et al., 15), (Shamir, 16), (Wang et al., 17)
  Communication, Computation, Memory efficient
  Highly synchronized (at each step, a different node performs stoch. updates)

Our work: decentralized (no special node; robust: no single point of overload and failure), implicit regularization (regularization path on the fly), first-order (low computational cost), iterative (early stopping), exploits concentration

Distributed Gradient Descent

- Agents maintain models $\omega_{t,v}$ and iteratively communicate (Nedic, Ozdaglar 09):

$$\omega_{t+1,v} = \underbrace{\sum_{w \in V} P_{vw}}_{\text{Aggregation}} \underbrace{\left(\omega_{t,w} - \eta_t \frac{1}{M} \sum_{i=1}^{M} \nabla \ell(\omega_{t,w}, Z_{i,w}) \right)}_{\text{Gradient Descent}}$$

- Implicit regularization (no constraints/penalization) allows:
  - **simple algorithm:** primal-only
  - **simple analysis:** cf. (Duchi et al., 12), dual method to avoid some technical difficulties introduced by the projection step and get dependence on $\frac{1}{1-\lambda_2}$
- **Proof of concept:** easy to add gossip acceleration (Chebyshev polynomial, lifted Markov Chain) and gradient acceleration (Nesterov, momentum)
- We treat algorithm as a **statistical device**

Implicit regularizers: time t, step size η_t (Single-Machine (Bauer et al., 2007, Yao et al., 2007)); matrix P (Richards and **R.**, 2018) arXiv:1809.06958

Nonparametric Regression

- Hilbert Space H. Supervised Learning: $Z = (X, Y)$, $X \subseteq H$, $Y \subseteq \mathbb{R}$
- **Square loss function** with **linear predictors**: $\ell(\omega, Z) = (\langle \omega, X \rangle_H - Y)^2$
- Optimal predictor $g^\star(x) := \mathbf{E}[Y|X = x]$
- Let $\overline{g}^\star$ denote projection of $g^\star$ onto **closure** of linear predictors
- "Extended covariance operator" from $L^2(H, \mathbf{P}(X \in \cdot))$ to $L^2(H, \mathbf{P}(X \in \cdot))$ $\mathcal{L}(g) = \mathbf{E}[\langle X, \cdot \rangle_H g(X)]$

Assumptions:

- There exists $\kappa \geq 1$ such that $\langle x, x' \rangle_H \leq \kappa^2 \quad \forall x, x' \in X$
- There exist $M > 0$ and $\nu > 1$ such that for any $\ell \in \mathbb{N}$ we have

$$\int_Y y^{2\ell} \mathbf{P}(Y \in dy|X = x) \leq \nu \ell! M^\ell \qquad \mathbf{P}(X \in \cdot)\text{-almost surely}$$

- **Source:** Complexity of optimal predictor through the covariance operator

  There exist $r > 0$ and $c > 0$ such that $\|\mathcal{L}^{-r}(\overline{g}^\star)\|_{L^2} \leq c$

  - Larger $r \Rightarrow$ easier problem; for $r \geq 1/2$, $\overline{g}^\star$ is in the space of functions searched
- **Capacity:** Decay of eigenvalue of the covariance operator

  For some $\gamma \in (0, 1]$, $\exists c$ such that $d_\lambda := \mathrm{Tr}(\mathcal{L}(\mathcal{L} + \lambda I)^{-1}) \leq c\lambda^{-\gamma} \; \forall \lambda \geq 0$

  - $\gamma = 1$, then always true
  - Smaller $\gamma \Rightarrow$ faster decay of eigenvalues and smaller effective dimension

Results

(Richards and R., 2019)

- Let $r \geq 1/2$ and $2r+\gamma > 2$. Let $M \geq N^{\frac{2r+\gamma+2}{2r+\gamma-2}}$, $M \geq N^{2r/\gamma}$.
- $t \simeq \underbrace{(NM)^{\frac{1}{2r+\gamma}}}_{\text{single-machine iterations (Lin and Rosasco, 2017)}} \times \left\{ \left(\frac{(NM)^{\frac{2r}{2r+\gamma}}}{M(1-\lambda_2)^\gamma} \right)^{1/\gamma} \vee 1 \right\}$

Then $\boxed{\sup_{v \in V} \mathbf{E}\, r(\omega_{t,v}) - r(\omega^\star) \lesssim (NM)^{-\frac{2r}{2r+\gamma}}}$

$M > N^{\frac{2r}{\gamma}} (\frac{1}{1-\lambda_2})^{2r+\gamma} \Rightarrow$ **Speedup** $= \frac{\cancel{\text{single-machine iterations}}}{\cancel{\text{decentralized iterations}}} \frac{NM}{\tau + \deg(P) + M} \overset{\text{if } M \gtrsim \tau + \deg(P)}{=} O(N)$

	Cycle	Grid	R. Geom.	Complete	Expander
Statistics†	optimal	optimal	optimal	optimal	optimal
Speedup	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$
Communication	$O(1)$	$O(1)$	$O(1)$	$O(N)^\ddagger$	$O(1)$

†(Caponetto, De Vito, 07) $r \geq 1/2, \gamma \in [0,1]$ $\ddagger$ If M large enough, use sparse commun.

Using statistics we can meet all three *desiderata* with a simple algorithm

Decentralized Optimization + Statistics: No Trade-Off

Consensus Optimization:

Graphs with smaller spectral gap benefit less from decentralization

Trade-off speedup and communication

Consensus Optimization + **Statistics**:

Regime ($M \gtrsim N^{\alpha}(\frac{1}{1-\lambda_2})^{\beta}$) where NO trade-off

Proof Sketch: Auxiliary Devices

- Full-batched gradient descent:

$$\omega_{t+1,v} = \sum_{w \in V} P_{vw} \left(\omega_{t,w} - \eta_t \frac{1}{M} \sum_{i=1}^{M} \left(\langle \omega_{t,w}, x_{i,w} \rangle_H - y_{i,w} \right) x_{i,w} \right)$$

- Two auxiliary devices:
  - "Population" gradient descent: (Lin and Rosasco, 2017)

$$\mu_{t+1} = \mu_t - \eta_t \int \left(\langle \mu_t, x \rangle_H - g^\star(x) \right) x \, \mathbf{P}(X \in dx)$$

  - "Complete graph" gradient descent ($P = \frac{1}{N} \mathbf{1}\mathbf{1}^\top$) (Nedic and Ozdaglar, 09)

$$\xi_{t+1} = \xi_t - \eta_t \frac{1}{NM} \sum_{w \in V} \sum_{i=1}^{M} \left(\langle \xi_t, x_{i,w} \rangle_H - y_{i,w} \right) x_{i,w}$$

(corresponds to single-machine gradient descent with all NM samples)

Proof Sketch: Error Decomposition

Define $\mathcal{S} : H \to L^2(H, \mathbf{P}(X \in \cdot))$ as $\mathcal{S}\omega = \langle \omega, \cdot \rangle_H$

$$r(\omega_{t,v}) - \inf_{\omega \in H} r(\omega) \leq \underbrace{2 \underbrace{\|\mathcal{S}\mu_t - \overline{g}^\star\|_{L^2}^2}_{\text{bias}} + 4 \underbrace{\|\mathcal{S}(\xi_t - \mu_t)\|_{L^2}^2}_{\text{sample variance}}}_{\text{single-machine gradient descent}} + 4 \underbrace{\|\mathcal{S}(\omega_{t,v} - \xi_t)\|_{L^2}^2}_{\text{network error}}$$

Proof: $r(\omega) - \inf_{\omega \in H} r(\omega) = \|\mathcal{S}\omega - \overline{g}^\star\|_{L^2}^2$ (Rosasco, Villa, 15); $(a+b)^2 \leq 2(a^2+b^2)$

- Analysis of single-machine gradient descent yields (Lin and Rosasco, 17)

$$\mathbf{E}[\textbf{single-machine g.d.}] \lesssim \underbrace{(\eta t)^{-2r}}_{\text{bias}} + \underbrace{(NM)^{-\frac{2r}{2r+\gamma}} (1 \vee (NM)^{-\frac{1}{2r+\gamma}} \eta t)^2}_{\text{sample variance}}$$

- Using **concentration**, analysis of network term yields, for $r \geq 1/2$,

$$\mathbf{E}[\textbf{network error}] \lesssim \frac{\eta^2}{M(1-\lambda_2)^2} + \frac{(\eta t)^{2+\gamma}}{M^2}$$

Proof Sketch: Network Error

- Population quantities:
  - Let $\mathcal{T} : H \to H$ defined as $\mathcal{T} := \int \langle \cdot, x \rangle_H \, x \, \mathbf{P}(X \in dx)$
  - Let $\Pi_{t:k+1} := (I - \eta_t \mathcal{T})(I - \eta_{t-1}\mathcal{T}) \cdots (I - \eta_{k+1}\mathcal{T})$
- Empirical quantities:
  - Let $\widehat{\mathcal{T}}_{\mathbf{x}_v} : H \to H$ defined as $\widehat{\mathcal{T}}_{\mathbf{x}_v} = \frac{1}{M} \sum_{i=1}^{M} \langle \cdot, x_{i,v} \rangle_H x_{i,v}$
  - Let $\widehat{\Pi}_{t:k+1} := (I - \eta_t \widehat{\mathcal{T}}_{\mathbf{x}_{w_t}})(I - \eta_{t-1}\widehat{\mathcal{T}}_{\mathbf{x}_{w_{t-1}}}) \cdots (I - \eta_{k+1}\widehat{\mathcal{T}}_{\mathbf{x}_{w_k}})$

Adding and subtracting the population process $\{\mu_k\}_{k \geq 0}$ (to exploit concentration)

$$\|\mathcal{S}(\omega_{t+1,v} - \xi_{t+1})\|_{L^2}^2 = \left\| \sum_{k=1}^{t} \eta_k \sum_{w_{t:k} \in V^{t-k+1}} \left(P_{v w_{t:k}} - \frac{1}{N^{t-k+1}} \right) \mathcal{T}^{1/2} \widehat{\Pi}_{t:k+1} \widehat{\varepsilon}_{k,w_k} \right\|_H^2$$

with $\widehat{\varepsilon}_{k,v} := (\mathcal{T}\mu_k - \mathcal{S}^\star g^\star) - (\widehat{\mathcal{T}}_{\mathbf{x}_v}\mu_k - \widehat{\mathcal{S}}^\star_{\mathbf{x}_v}\mathbf{y}_v)$ where

- $\mathcal{S}^\star : L^2(H, \mathbf{P}(X \in \cdot)) \to H$ adjoint operator of $\mathcal{S}$: $\mathcal{S}^\star g = \int x g(x) \mathbf{P}(X \in dx)$
- $\widehat{\mathcal{S}}_{\mathbf{x}_v} : H \to \mathbb{R}^M$ defined as $(\widehat{\mathcal{S}}_{\mathbf{x}_v}\omega)_i = \langle \omega, x_{i,v} \rangle_H$
- $\widehat{\mathcal{S}}^\star_{\mathbf{x}_v} : \mathbb{R}^M \to H$ adjoint operator of $\widehat{\mathcal{S}}_{\mathbf{x}_v}$: $\widehat{\mathcal{S}}^\star_{\mathbf{x}_v}\mathbf{y}_v = \frac{1}{M} \sum_{i=1}^{M} y_{i,v} x_{i,v}$

Key properties: $\mathbf{E}\,\widehat{\varepsilon}_{k,v} = 0$ and $\widehat{\varepsilon}_{k,v}$ concentrates

Proof Sketch: Network Error using Concentration

Adding and subtracting population covariance operators:

$$\|\mathcal{S}(\omega_{t+1,v} - \xi_{t+1})\|_{L^2}^2 \leq \underbrace{2\left\| \sum_{k=1}^{t} \eta_k \sum_{w_{t:k} \in V^{t-k+1}} \left(P_{v w_{t:k}} - \frac{1}{N^{t-k+1}} \right) \mathcal{T}^{1/2} \Pi_{t:k+1} \widehat{\varepsilon}_{k,w_k} \right\|_H^2}_{\text{population covariance error}}$$

$$+ \underbrace{2\left\| \sum_{k=1}^{t} \eta_k \sum_{w_{t:k} \in V^{t-k+1}} \left(P_{v w_{t:k}} - \frac{1}{N^{t-k+1}} \right) \mathcal{T}^{1/2} (\widehat{\Pi}_{t:k+1} - \Pi_{t:k+1}) \widehat{\varepsilon}_{k,w_k} \right\|_H^2}_{\text{residual empirical covariance error}}$$

Concentration in Hilbert spaces (Pinelis and Sakhanenko,1986), (Caponnetto and De Vito, 2007), (Lin and Rosasco, 2017): for any $r \geq 1/2$, $\lambda > 0$, w.p.a.l. $1 - \delta$

$$\|(\mathcal{T} + \lambda)^{-1/2} \widehat{\varepsilon}_{k,v}\|_H^2 \lesssim \frac{1}{M\lambda} \log^2 \frac{1}{\delta}$$

$$\|(\mathcal{T} + \lambda)^{-1/2} (\widehat{\mathcal{T}}_{\mathbf{x}_v} - \mathcal{T})\|_{HS}^2 \lesssim \frac{1}{M\lambda} \log^2 \frac{1}{\delta}$$

$$\boxed{\mathbf{E}[\text{network error}] \lesssim \frac{\eta^2}{M(1-\lambda_2)^2} + \frac{(\eta t)^{2+\gamma}}{M^2}}$$

Online Learning (Multi-Armed Bandit)

(joint work D. Martínez-Rubio and V. Kanade)

Single-Machine Learning

$Z_1, \ldots, Z_{TN} \sim \mathbf{P}$

($\mathbf{P}$ unknown)

Goal

Minimize expected risk: $r(\omega) = \mathbf{E}\,\ell(\omega, Z)$

Online Learning (Multi-Armed Bandit)

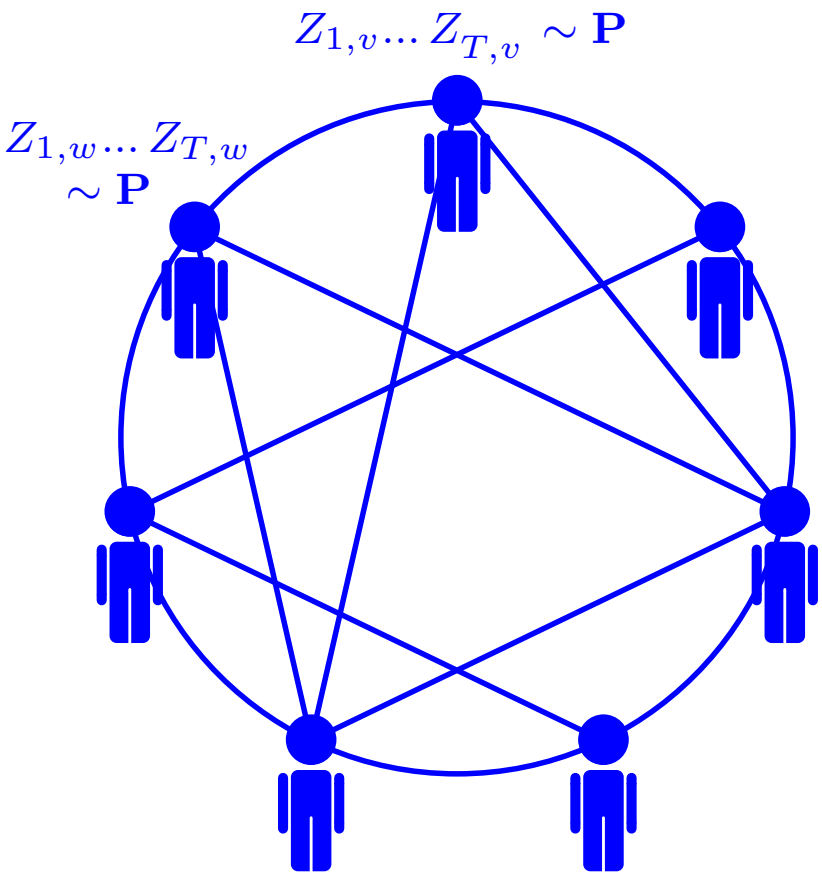
Data arrive sequentially, partially observed

- Play action $\omega_t \in [K]$
- Data $Z_t = \begin{bmatrix} Z_t^1 \\ \vdots \\ Z_t^K \end{bmatrix}$ indep. components
- $\ell(\omega_t, Z_t) = -Z_t^{\omega_t}$ (observe component ω_t)

Regret: $\displaystyle\sum_{t=1}^{TN} r(\omega_t) - TN \inf_{\omega} r(\omega)$

$$TN \max_{\omega \in [K]} \mu^\omega - \sum_{t=1}^{TN} \mu^{\omega_t} \quad (\mu^\omega = r(\omega) = \mathbf{E}Z^\omega)$$

Decentralized Learning



Goal

Minimize expected risk: $r(\omega) = \mathbf{E}\,\ell(\omega, Z)$

Online Learning (Multi-Armed Bandit)

Data arrive sequentially, partially observed

- Play action $\omega_{t,v} \in [K]$
- Data $Z_{t,v} = \begin{bmatrix} Z_{t,v}^1 \\ \vdots \\ Z_{t,v}^K \end{bmatrix}$ indep. components
- $\ell(\omega_{t,v}, Z_{t,v}) = -Z_{t,v}^{\omega_{t,v}}$

Regret: $\displaystyle\sum_{v \in V} \sum_{t=1}^{T} r(\omega_{t,v}) - TN \inf_{\omega} r(\omega)$

Q. Interplay statistics, computation, communication?

Single-Machine UCB

approx. held by agent v

- n_t^k: number of times arm k is pulled up to time t $\rightarrow$ $n_{t,v}^k$
- m_t^k: cumulative rewards from arm k up to time t $\rightarrow$ $m_{t,v}^k$

Upper confidence bound: $U_t^k = \frac{m_t^k}{n_t^k} + \sqrt{\frac{\eta \log t}{n_t^k}}$ $\rightarrow$ $U_{t,v}^k$

Algorithm: UCB

---

for $t = 1, \ldots, K$ **do**
  play arm $\omega_t = t$ $\rightarrow$ $\omega_{t,v} = t$
end
for $t = K+1, \ldots, T$ **do**
  play arm $\omega_t \in \operatorname{argmax}_{k \in [K]} U_{t-1}^k \rightarrow$ { computation: $\omega_{t,v} \in \operatorname{argmax}_{k \in [K]} U_{t-1,v}^k$; communication: $n_{t,v}^k, \mu_{t,v}^k$
end

Q: Which protocol should we use to compute $n_{t,v}^k$ and $n_{t,v}^k$?

Decentralized UCB

Running Consensus (Braca et al., 2008)

- Values sequentially added to the network $y_1, \dots, y_t \in \mathbb{R}^N$ ($2K$ of these)
- Running consensus approximation: $x_{t+1} = Px_t + y_t$ with $x_1 = 0$

Approach used in (Landgren et al., 2016):

$$\mathbf{E}[\text{Regret}] \lesssim \sum_{k:\Delta_k>0} \sum_{j=1}^{N} \frac{8\gamma(1+\epsilon_c^j)}{N\Delta_k} \ln(T) + N\left(\frac{\gamma}{\gamma-1} + \sqrt{N}\sum_{j=2}^{N}\frac{\lambda_j}{1-\lambda_j}\right)\sum_{k=1}^{K}\Delta_k$$

Problem: Poor approximation due to last-added values not sufficiently mixed

$$x_{t+1} = \underbrace{y_t + Py_{t-1} + \cdots}_{\text{NOT well-mixed}} + \underbrace{\cdots + P^{t-2}y_2 + P^{t-1}y_1}_{\text{well-mixed}}$$

IDEA: Take decisions only based on well-mixed (delayed) information

- Use mixing to prescribe delay
- Delay does not increase regret much: τ-delay increases regret by $\tau \sum_k \Delta_k$ (Joulani et al., 2013)

Delayed Decentralized UCB

IDEA: Graph-dependent delay to make decision using well-mixed information:

$$C = \left\lceil \frac{\ln(2N/\epsilon)}{\sqrt{2(1-\lambda_2)}} \right\rceil \quad \text{yields} \quad \left\| p_C(P)y - \frac{\mathbf{1}\mathbf{1}^\top y}{N} \right\|_2 \leq \frac{\epsilon}{N}$$

Algorithm: Delayed Decentralized UCB

---

for $t = 1, \ldots, K$ **do** play arm $\omega_{t,v} = t$ **end**

for *Rounds of* C *time steps until* $t = T$ **do**

Computation:
- ① compute optimal arm using local aggregate statistics
  $\omega_{\text{current round},v} \in \operatorname{argmax}_{k\in[K]} U^k_{\text{rounds up to previous},v}$
- ② play optimal arm for C times and store information
  $z_{\text{current round}} = y_t + y_{t-1} + \cdots + y_{t-C+1}$

Communication:
- ③ mix information stored on previous round for C steps
  $p_C(P) z_{\text{previous round}}$
- ④ add mix information to aggregate local statistics
  $\sum_{\text{rounds up to previous}} p_C(P) z_{\text{round}}$

end

Results

DDUCB (Martínez-Rubio, Kanade, R., 2019)

$$\mathbf{E}[\text{Regret}] \lesssim \sum_{k:\Delta_k>0} \frac{\sigma^2 \ln(TN)}{\Delta_k} + \frac{N \ln(N)}{\sqrt{1-\lambda_2}} \sum_{k=1}^{K} \Delta_k$$

- Running N separate instances of UCB at each node (no communication):

$$O\left(\sum_{k:\Delta_k>0} \frac{N \ln(T)}{\Delta_k} + N \sum_{k=1}^{K} \Delta_k \right)$$

- Running single-machine UCB algorithm for TN steps (no batches):

$$O\left(\sum_{k:\Delta_k>0} \frac{\ln(TN)}{\Delta_k} + \sum_{k=1}^{K} \Delta_k \right)$$

- Lower-bound (immediate): $\Omega\left(\sum_{k:\Delta_k>0} \frac{\sigma^2 \ln(TN)}{\Delta_k} + \left(\frac{N}{K} + 1 \right) \sum_{k=1}^{K} \Delta_k \right)$

  Dependence on N is natural from batched bandits (Perchet et al, 2016)

Results

DDUCB (Martínez-Rubio, Kanade, R., 2019)

$$\mathbf{E}[\text{Regret}] \lesssim \sum_{k:\Delta_k>0} \frac{\sigma^2 \ln(TN)}{\Delta_k} + \frac{N \ln(N)}{\sqrt{1-\lambda_2}} \sum_{k=1}^{K} \Delta_k$$

- **Statistics:** Do we recover optimal guarantees w.r.t. NT samples?
  - First term regret: **Yes** (up to $\log N$ factor)
  - Second term regret: **Up to a factor** $\frac{\log N}{\sqrt{1-\lambda_2}}$
- **Speedup** $= \dfrac{N}{1+\tau+\deg(P)} \sim N$ if $\deg(P)$ is constant
- **Communication:** $O(\deg(P) \cdot K)$

Can achieve optimal regret by adding communication rounds (multi-step)

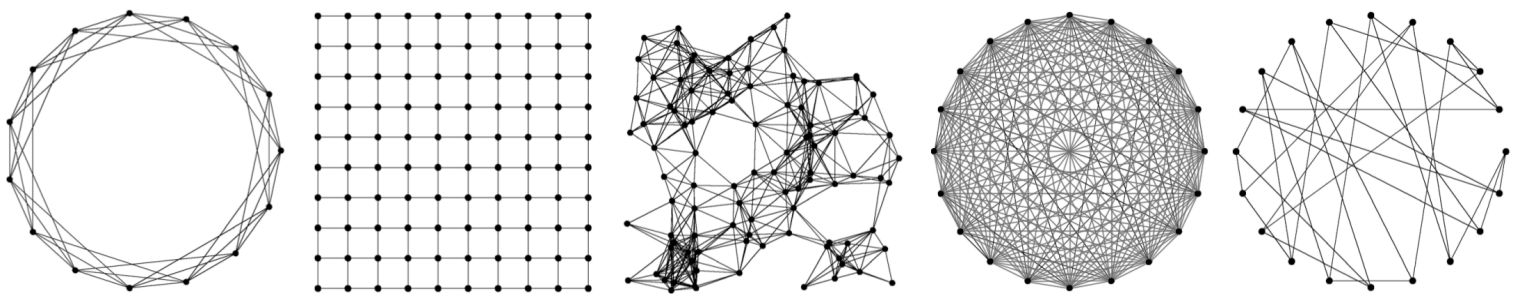
Trade-off between statistics and computation/communication

Results

DDUCB (Martínez-Rubio, Kanade, R., 2019)

$$\mathbf{E}[\text{Regret}] \lesssim \alpha \sum_{k:\Delta_k>0} \frac{\sigma^2 \ln(TN)}{\Delta_k} + \beta N \log(N) \sum_{k=1}^{K} \Delta_k$$

		Cycle	Grid	R. Geom.	Complete	Expander
Statistics	α	1	1	1	1	1
	β	N	$\sqrt{N}$	$\sqrt{N}$	1	1
Speedup		$O(N)$	$O(N)$	$O(N)$	$O(1)$	$O(N)$
Communication		$O(1)$	$O(1)$	$O(1)$	$O(N)$	$O(1)$



Trade-Off Statistics Speedup and Communication

Decentralized Bandits:

Graphs with smaller spectral gap benefit less from decentralization

Trade-off between statistics, speedup and communication

Remark: it is easy to be statistically optimal (i.e., to achieve the network regret for batched bandits) by adding extra communication rounds in between agents' decisions, like multi-step methods for consensus optimization

Experiments

Graph: Cycle	DDUCB	coopUCB (Landgren et al., 16)
α	$O(1)$	$\Theta(N^2 \log(N))$
β	$\Theta(N^2)$	$\Theta(N^{7/2})$

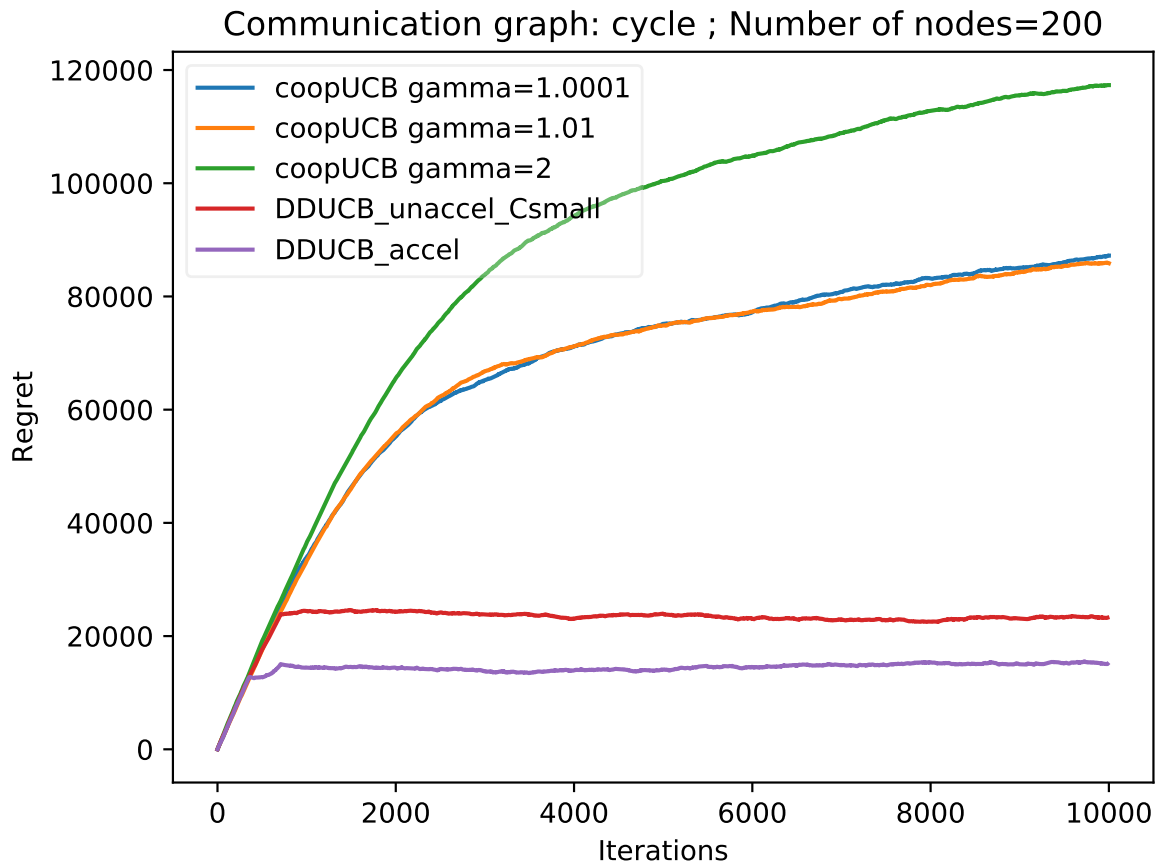


Figure: $K = 17$; optimal arm $\mathcal{N}(1, 1)$; sub-optimal arms $\mathcal{N}(0.8, 1)$

Proof Sketch

- Network Regret $= \sum_{k=1}^{K} \Delta_k \mathbf{E}\, n_T^k$ (Aurer er al., 2002)

- **Relate agents' approximations ($n_{t,v}^k$) to network's quantities (n_t^k):**
  - Variables used by agents to make decisions, based on delayed information:

$$\frac{n_{t,v}^k}{N} = \sum_{\substack{\ell \in \{\text{pulls from arm } k \\ \text{up to delayed}(t)\}}} \text{weight}_\ell \qquad \frac{m_{t,v}^k}{N} = \sum_{\substack{\ell \in \{\text{pulls from arm } k \\ \text{up to delayed}(t)\}}} \text{weight}_\ell \cdot \text{reward}_\ell$$

  - $\text{weight}_\ell = p_C(P)_{vw}$ for some $w \in V$, $\text{delayed}(t) = K + \lfloor \frac{t-K}{C} \rfloor C - C$
  - $\text{delay}(t) = t - \text{delayed}(t) \in [C, 2C]$
  - Accelerated mixing yields $|\text{weight}_\ell - \frac{1}{N}| \leq \frac{\varepsilon}{N}$
  - This yields $n_{t,v}^k \geq (1-\varepsilon) n^k_{\text{delayed}(t)}$
  - Delay does not increase regret by much: τ-delay yields increase $\tau \sum_k \Delta_k$

- **Control prob. of playing sub-optimal arm given agents' information:**
  - Control on weights yields $\mathbf{P}(|\frac{m_{t,v}^k}{n_{t,v}^k} - \mu_i| \geq \sqrt{\frac{2\eta \log t}{n_{t,v}^k}}) \leq \frac{2}{t^{(\eta+1)/2}}$
  - This yields $\mathbf{P}(\omega_{t,v} = k | n_{t,v}^k > \frac{8\eta \log s}{\Delta_k^2}) \leq \frac{4}{s^{(\eta+1)/2}}$

Conclusion

Investigated interplay between **statistics**, **computation**, **communication**

Offline learning: (arXiv:1905.03135 and also arXiv:1809.06958)

- Implicit regularization for Distributed GD (simple method; proof of concept)
- Use statistics: exploit concentration of local objectives
- **No trade-off:** regime ($M \gtrsim N^{\alpha}(\frac{1}{1-\lambda_2})^{\beta}$) where all *desiderata* are met

Online learning: (arXiv:1810.04468)

- New algorithm based on graph-dependent mixing delay
- Graph-independent $\log T$-term in regret
- **Trade-off** between statistics and computation/communication