Towards a Better Theoretical Understanding of Independent Subnetwork Training

Egor Shulgin **Peter Richtárik**

King Abdullah University of Science and Technology (KAUST)
Thuwal, Saudi Arabia

ICML 2023 Workshop on Federated Learning and Analytics in Practice
July 2023

King Abdullah University
of Science and Technology

Problem formulation

$$\min_{x \in \mathbb{R}^d} \left[f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right] \tag{1}$$

- n is the number of **workers**
- each $f_i : \mathbb{R}^d \to \mathbb{R}$ represents the **loss** of the model
- parameterized by vector $x \in \mathbb{R}^d$ on the data of client i

Problem formulation

$$\min_{x \in \mathbb{R}^d} \left[f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right] \tag{1}$$

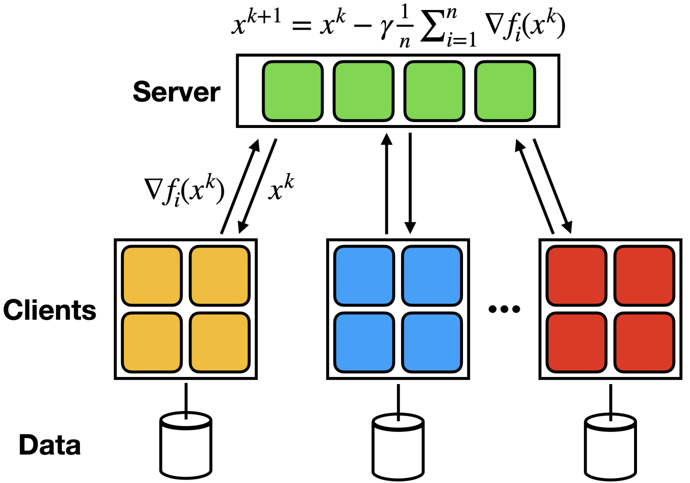
- n is the number of **workers**
- each $f_i : \mathbb{R}^d \to \mathbb{R}$ represents the **loss** of the model
- parameterized by vector $x \in \mathbb{R}^d$ on the data of client i

Typical (Stochastic) Gradient Descent-type method for solving problem (1):

$$x^{k+1} = x^k - \gamma g^k, \qquad g^k = \frac{1}{n} \sum_{i=1}^{n} g_i^k \tag{2}$$

- $\gamma > 0$ is the **stepsize**
- g_i^k is a suitably constructed **estimator** of $\nabla f_i(x^k)$

Standard distributed learning setting $\frac{1}{n}\sum_{i=1}^{n} f_i(x) \to \min_{x\in\mathbb{R}^d}$

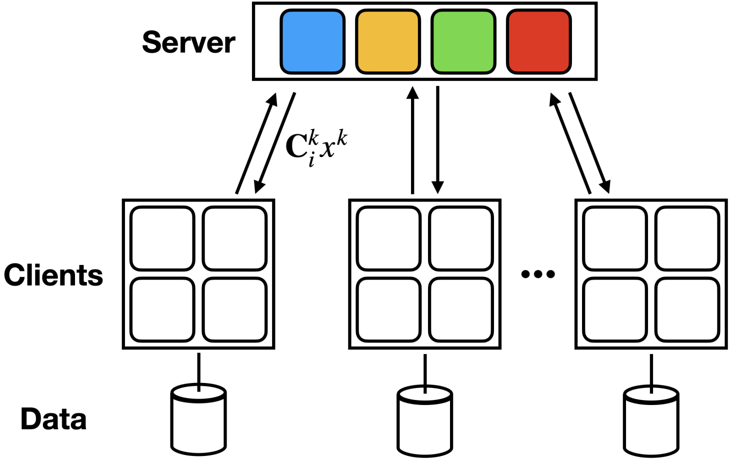


Distributed Gradient Descent architecture example (based on Dean et al., 2012)

Allows to employ **data parallelism** to speed up training.

Data + Model parallelism

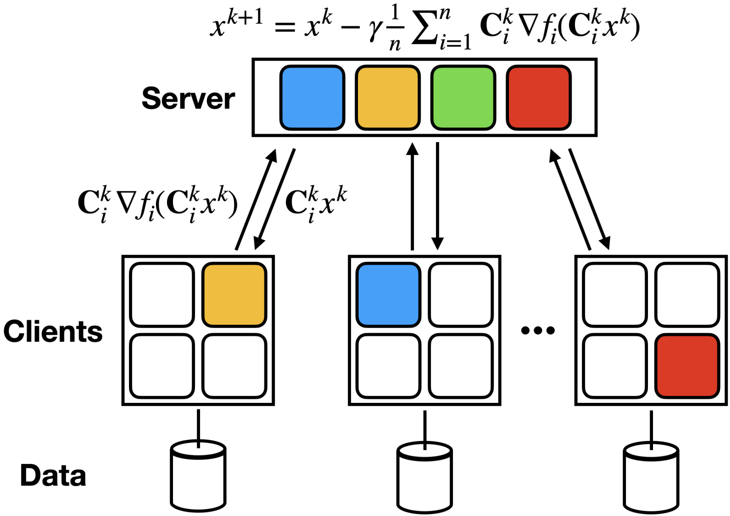
$$x^{k+1} = x^k - \gamma \frac{1}{n} \sum_{i=1}^{n} \mathbf{C}_i^k \nabla f_i(\mathbf{C}_i^k x^k)$$



Distributed Gradient Descent with sparse models

1. Sample parameters / Decompose the model

Data + Model parallelism

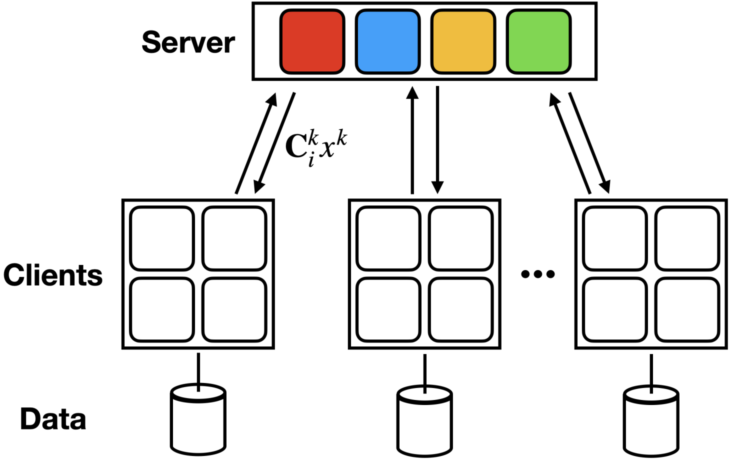


Distributed Gradient Descent with sparse models

2. Perform local computations w.r.t. submodels

Data + Model parallelism

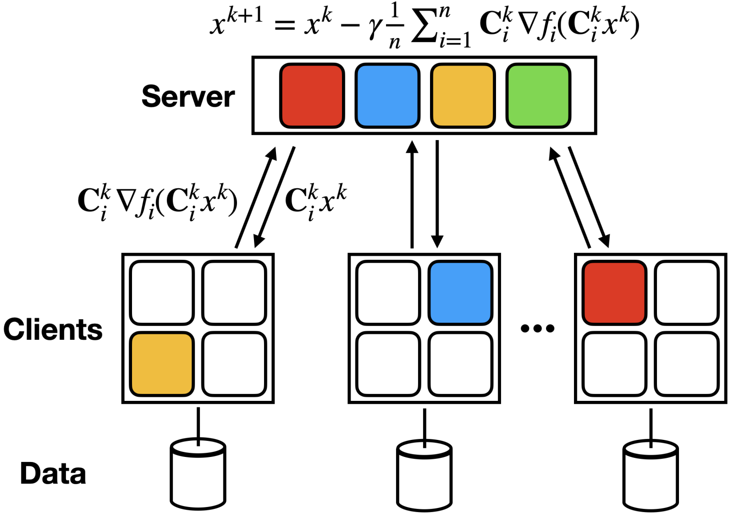
$$x^{k+1} = x^k - \gamma \frac{1}{n} \sum_{i=1}^{n} \mathbf{C}_i^k \nabla f_i(\mathbf{C}_i^k x^k)$$



Distributed Gradient Descent with sparse models

3. Sample **new** parameters / Decompose the model

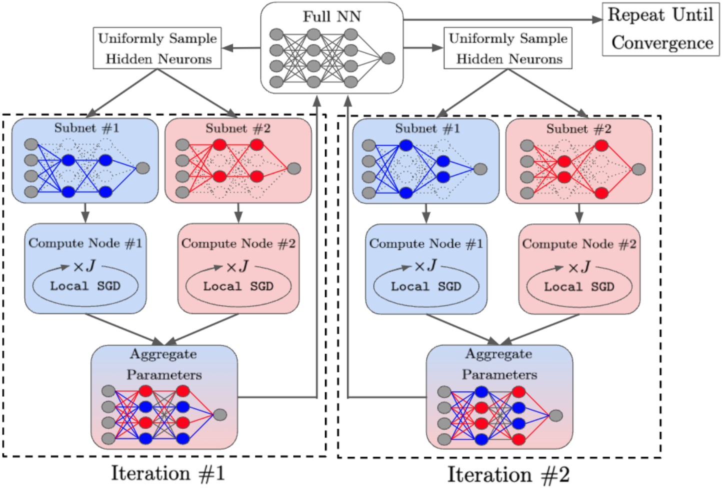
Data + Model parallelism



Distributed Gradient Descent with sparse models

4. Perform local computations w.r.t. **new** submodels

Independent Subnetwork Training (IST) [Yuan et al., 2022]

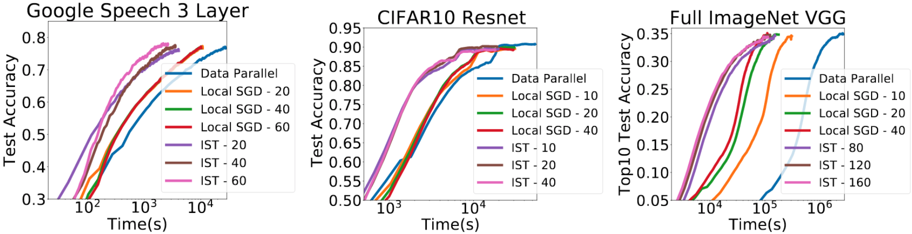


Schematic depiction of a NN trained with IST across two nodes (source: Yuan et al., 2022)

Efficiently combines **data** and **model parallelism**.

Brief history of IST

- Originally suggested in 2019 by Yuan et al. (2022) for networks with **fully connected** layers.
- Later extended to **ResNets** (Dun et al., 2022) and **Graph architectures** (Wolfe et al., 2021).
- Analyzed for overparameterized **single hidden layer** NNs with ReLU activations (Liao and Kyrillidis, 2022).
- Expanded to the **federated setting** via an asynchronous distributed dropout technique (Dun et al., 2023).



IST showed impressive **empirical performance** (source: Yuan et al., 2022)

Modeling IST via sketching

Submodel computations can be represented by using sketches

$$g_i^k := \mathbf{C}_i^k \nabla f_i(\mathbf{C}_i^k x^k), \tag{3}$$

for symmetric positive semi-definite **matrices** $\mathbf{C}_i^k \in \mathbb{R}^{d\times d}$ (e.g. $\mathbf{C}_i = e_i e_i^\top$, e_i – basis vectors). Then IST (with 1 GD step) can be modeled as

$$x^{k+1} = \frac{1}{n}\sum_{i=1}^{n}\left[\mathbf{C}_i^k x^k - \gamma \mathbf{C}_i^k \nabla f_i(\mathbf{C}_i^k x^k)\right]. \tag{4}$$

Modeling IST via sketching

Submodel computations can be represented by using sketches

$$g_i^k := \mathbf{C}_i^k \nabla f_i(\mathbf{C}_i^k x^k), \tag{3}$$

for symmetric positive semi-definite **matrices** $\mathbf{C}_i^k \in \mathbb{R}^{d\times d}$ (e.g. $\mathbf{C}_i = e_i e_i^\top$, e_i – basis vectors). Then IST (with 1 GD step) can be modeled as

$$x^{k+1} = \frac{1}{n}\sum_{i=1}^{n}\left[\mathbf{C}_i^k x^k - \gamma \mathbf{C}_i^k \nabla f_i(\mathbf{C}_i^k x^k)\right]. \tag{4}$$

Permutation Sketch (for $n = d$) [Szlendak, Tyurin, and Richtárik, 2022]

Let $\pi = (\pi_1, \ldots, \pi_d)$ be a random **permutation** of $[d] := (1, \ldots, d)$. Then for each $i \in [n]$, define `Perm-q` operator

$$\mathbf{C}_i := n \cdot \sum_{j=q(i-1)+1}^{qi} e_{\pi_j} e_{\pi_j}^\top. \tag{5}$$

Challenges in analysis

Gradient estimator is **biased** even if $\mathbf{C}$ is unbiased unlike for Compressed Gradient Descent-type methods

$$\mathbb{E}\left[\nabla f(\mathbf{C}x)\right] \neq \nabla f(x) = \mathbb{E}\left[\mathbf{C}\nabla f(x)\right] = \mathbb{E}\left[\mathbf{C}\right] \nabla f(x). \tag{6}$$

Challenges in analysis

Gradient estimator is **biased** even if $\mathbf{C}$ is unbiased unlike for Compressed Gradient Descent-type methods

$$\mathbb{E}\left[\nabla f(\mathbf{C}x)\right] \neq \nabla f(x) = \mathbb{E}\left[\mathbf{C}\nabla f(x)\right] = \mathbb{E}\left[\mathbf{C}\right] \nabla f(x). \tag{6}$$

Previous works rely on **bounded** expected stochastic **gradient norm**:

$$\mathbb{E}\left[\left\|\nabla f(\mathbf{C}x)\right\|^2\right] \leq G \tag{7}$$

and may not hold, even for quadratic functions

$$f(x) = x^\top \mathbf{A} x, \tag{8}$$

as $\|\nabla f(x)\| = \|\mathbf{A}x\|$ is unbounded for $x \in \mathbb{R}^d$.

Simplifications taken

1. Every node i computes **full gradient** at the submodel $\mathbf{C}_i \nabla f_i(\mathbf{C}_i x^k)$
2. Nodes perform **one descent** step (or just gradient computation)
3. Special case of a convex symmetric **quadratic model** as a loss function

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x), \qquad f_i(x) \equiv \frac{1}{2} x^\top \mathbf{L}_i x - x^\top \mathrm{b}_i \,. \tag{9}$$

Simplifications taken

1. Every node i computes **full gradient** at the submodel $\mathbf{C}_i \nabla f_i(\mathbf{C}_i x^k)$
2. Nodes perform **one descent** step (or just gradient computation)
3. Special case of a convex symmetric **quadratic model** as a loss function

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \qquad f_i(x) \equiv \frac{1}{2} x^\top \mathbf{L}_i x - x^\top \mathrm{b}_i\,. \tag{9}$$

In this instance, the **gradient estimator** takes the form

$$g^k = \frac{1}{n}\sum_{i=1}^{n} g_i^k = \frac{1}{n}\sum_{i=1}^{n} \mathbf{C}_i^k \left(\mathbf{L}_i \mathbf{C}_i^k x^k - \mathrm{b}_i\right) = \boxed{\overline{\mathbf{B}}^k x^k - \overline{\mathbf{C}\mathrm{b}}}, \tag{10}$$

where $\overline{\mathbf{B}}^k := \frac{1}{n}\sum_{i=1}^{n} \mathbf{C}_i^k \mathbf{L}_i \mathbf{C}_i^k$ and $\overline{\mathbf{C}\mathrm{b}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{C}_i^k\, \mathrm{b}_i$.

Preconditioned permutation sparsification

Gradient estimator reminder

$$g^k = \frac{1}{n}\sum_{i=1}^{n} \mathbf{C}_i^k \mathbf{L}_i \mathbf{C}_i^k x^k - \frac{1}{n}\sum_{i=1}^{n} \mathbf{C}_i^k \mathrm{b}_i = \overline{\mathbf{B}}^k x^k - \overline{\mathbf{C}\mathrm{b}}\,. \tag{11}$$

Perm-1 modification

$$\tilde{\mathbf{C}}_i := \sqrt{n/\left[\mathbf{L}_i\right]_{\pi_i,\pi_i}}\, e_{\pi_i} e_{\pi_i}^\top. \tag{12}$$

Preconditioned permutation sparsification

Gradient estimator reminder

$$g^k = \frac{1}{n}\sum_{i=1}^{n} \mathbf{C}_i^k \mathbf{L}_i \mathbf{C}_i^k x^k - \frac{1}{n}\sum_{i=1}^{n} \mathbf{C}_i^k \mathrm{b}_i = \overline{\mathbf{B}}^k x^k - \overline{\mathbf{C}\mathrm{b}}\,. \tag{11}$$

Perm-1 modification

$$\tilde{\mathbf{C}}_i := \sqrt{n/\left[\mathbf{L}_i\right]_{\pi_i,\pi_i}}\, e_{\pi_i} e_{\pi_i}^\top . \tag{12}$$

In this case

$$\mathbb{E}\left[\tilde{\mathbf{C}}_i \mathbf{L}_i \tilde{\mathbf{C}}_i\right] = \mathbf{I}, \qquad \mathbb{E}\left[\overline{\mathbf{B}}^k\right] = \mathbf{I} \tag{13}$$

and

$$\mathbb{E}\left[\overline{\mathbf{C}\mathrm{b}}\right] = \frac{1}{\sqrt{n}}\frac{1}{n}\sum_{i=1}^{n} \mathbf{D}_i^{-\frac{1}{2}}\, \mathrm{b}_i\,. \tag{14}$$

The resulting gradient estimator

$$g^k = \overline{\mathbf{B}}^k \, x^k - \overline{\mathbf{C}\mathrm{b}} \tag{15}$$

Combined with modified preconditioned `Perm-1`

$$\begin{aligned}
\mathbb{E}\left[g^k\right] &= x^k - \frac{1}{\sqrt{n}}\frac{1}{n}\sum_{i=1}^{n} \mathbf{D}_i^{-\frac{1}{2}} \, \mathrm{b}_i & (16)\\
&= \overline{\mathbf{L}}^{-1} \nabla f(x^k) + \underbrace{\overline{\mathbf{L}}^{-1} \, \overline{\mathrm{b}} - \frac{1}{\sqrt{n}} \widetilde{\mathbf{D}\,\mathrm{b}}}_{h}, & (17)
\end{aligned}$$

where $\widetilde{\mathbf{D}\,\mathrm{b}} \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathbf{D}_i^{-\frac{1}{2}} \, \mathrm{b}_i$.

One main result

Convergence of **IST** to neighborhood

Assume that for every $\mathbf{D}_i := \mathrm{Diag}(\mathbf{L}_i)$ matrices $\mathbf{D}_i^{-\frac{1}{2}}$ exist, and heterogeneity is bounded as

$$\mathbb{E}\left[\left\|g^k - \mathbb{E}\left[g^k\right]\right\|^2_{\overline{\mathbf{L}}}\right] \leq \sigma^2. \tag{18}$$

Then, for the step size chosen as $0 < \gamma \leq \frac{1/2-\beta}{\beta+1/2}$, for $\beta \in (0, 1/2)$, the iterates of IST satisfy

$$\begin{aligned} \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2_{\overline{\mathbf{L}}^{-1}}\right] \quad &\leq \quad \frac{2\left(f(x^0) - \mathbb{E}\left[f(x^K)\right]\right)}{\gamma K} \\ &\quad + \left(2\beta^{-1}(1-\gamma) + \gamma\right)\|h\|^2_{\overline{\mathbf{L}}} + \gamma\sigma^2. \end{aligned} \tag{19}$$

Limitations of prior works

Originally Yuan et al. (2022) performed convergence analysis using the framework of GD with compressed iterates (Khaled and Richtárik, 2019).

- **Setting:** single-node stochastic case
  $\Rightarrow$ heterogeneity effect not captured.
- **Assumption** on sparsification parameter q:

$$\frac{d}{q} - 1 \lesssim \kappa^{-2} \qquad \Rightarrow \qquad \boxed{q \approx d} \tag{20}$$

- **Assumption** of Lipschitz continuity, which implies "bounded gradient"

$$\|\nabla f(x)\|^2 \leq G \tag{21}$$

Notation: κ – analogue for condition number of the optimized function.

Conclusions and future work

Takeaways

- It is possible to **precisely analyze** IST in a simplified setting.
- Even for quadratics naive IST **may not converge** to exact solution.

Conclusions and future work

Takeaways

- It is possible to **precisely analyze** IST in a simplified setting.
- Even for quadratics naive IST **may not converge** to exact solution.

Future work

- Extensions to settings like cross-device **federated learning**.
- Generalizations to **non-quadratics**.
- **Algorithmic modifications** of the original IST.

For more details, please refer to the paper arXiv:2306.16484

Any questions?

References

Dean, Jeffrey et al. (2012). "Large Scale Distributed Deep Networks". In: *Advances in Neural Information Processing Systems* 25.

Dun, Chen et al. (2022). "ResIST: Layer-wise decomposition of ResNets for distributed training". In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 610–620.

Dun, Chen et al. (2023). "Efficient and Light-Weight Federated Learning via Asynchronous Distributed Dropout". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 6630–6660.

Khaled, Ahmed and Peter Richtárik (2019). "Gradient descent with compressed iterates". In: *arXiv preprint arXiv:1909.04716*.

Liao, Fangshuo and Anastasios Kyrillidis (2022). "On the Convergence of Shallow Neural Network Training with Randomly Masked Neurons". In: *Transactions on Machine Learning Research*. URL: `https://openreview.net/forum?id=e7mYYMSyZH`.

Szlendak, Rafał, Alexander Tyurin, and Peter Richtárik (2022). "Permutation Compressors for Provably Faster Distributed Nonconvex Optimization". In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=GugZ5DzzAu`.

Wolfe, Cameron R et al. (2021). "GIST: Distributed training for large-scale graph convolutional networks". In: *arXiv preprint arXiv:2102.10424*.

Yuan, Binhang et al. (2022). "Distributed learning of fully connected neural networks using independent subnet training". In: *Proceedings of the VLDB Endowment* 15.8, pp. 1581–1590.

Algorithm Description (Supplementary Slide 1)

Algorithm 1 Distributed Submodel (Stochastic) Gradient Descent

```
Parameters: step size $\gamma > 0$; sketches $\mathbf{C}_1, \ldots, \mathbf{C}_n$; model $x^0 \in \mathbb{R}^d$
for $k = 0, 1, 2 \ldots$ do
    Select submodels $w_i^k = \mathbf{C}_i^k x^k$ for $i \in [n]$ and broadcast to all nodes
    for $i = 1, \ldots, n$ in parallel do
        Compute local (stochastic) gradient w.r.t. submodel: $\mathbf{C}_i^k \nabla f_i(w_i^k)$
        Take (multiple) gradient descent step $z_i^+ = w_i^k - \gamma \mathbf{C}_i^k \nabla f_i(w_i^k)$
        Send $z_i^+$ to the server
    end for
    Aggregate/merge received submodels: $x^{k+1} = \frac{1}{n} \sum_{i=1}^{n} z_i^+$
end for
```

Results in the interpolation case: $\mathrm{b}_i = 0$

Denote $\overline{\mathbf{L}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{L}_i \succ 0$.

Stationary point convergence for **general** sketches

If

$$\overline{\mathbf{W}} := \frac{1}{2}\mathbb{E}\left[\overline{\mathbf{L}}\,\overline{\mathbf{B}}^k + \overline{\mathbf{B}}^k\,\overline{\mathbf{L}}\right] \succeq 0, \tag{22}$$

and there exists a constant $\theta > 0$:

$$\mathbb{E}\left[\overline{\mathbf{B}}^k\,\overline{\mathbf{L}}\,\overline{\mathbf{B}}^k\right] \preceq \theta\,\overline{\mathbf{W}}, \tag{23}$$

and the step size is chosen as $0 < \gamma \leq \frac{1}{\theta}$, the iterates satisfy

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2_{\overline{\mathbf{L}}^{-1}\,\overline{\mathbf{W}}\,\overline{\mathbf{L}}^{-1}}\right] \leq \frac{2\left(f(x^0) - \mathbb{E}\left[f(x^K)\right]\right)}{\gamma K}. \tag{24}$$

Special case I: Gradient Descent

Consider $\mathbf{C}_i \equiv \mathbf{I}$. Then $\overline{\mathbf{B}}^k = \overline{\mathbf{L}}$ and for step size

$$\gamma = 1/\lambda_{\max}(\overline{\mathbf{L}}) \tag{25}$$

the iterates satisfy

$$\frac{1}{K}\sum_{k=0}^{K-1}\left\|\nabla f(x^k)\right\|_{\mathbf{I}}^2 \le \frac{2\lambda_{\max}(\overline{\mathbf{L}})\left(f(x^0)-f(x^K)\right)}{K}, \tag{26}$$

which matches $\mathcal{O}(1/K)$ rate of Gradient Descent in the non-convex setting.

Special case II: IST as `Perm-1`

Consider $\mathbf{C}_i^k = n e_{\pi_i^k} e_{\pi_i^k}^\top$. Then $\mathbb{E}\left[\mathbf{C}_i^k \mathbf{L}_i \mathbf{C}_i^k\right] = n \mathrm{Diag}(\mathbf{L}_i)$ and

$$\mathbb{E}\left[\overline{\mathbf{B}}^k\right] = \frac{1}{n}\sum_{i=1}^{n} n\mathrm{Diag}(\mathbf{L}_i) = \sum_{i=1}^{n} \mathbf{D}_i = n\,\overline{\mathbf{D}}\,\mathbf{1}. \tag{27}$$

Then inequality $\mathbb{E}\left[\overline{\mathbf{B}}^k\,\overline{\mathbf{L}}\,\overline{\mathbf{B}}^k\right] \preceq \theta\,\overline{\mathbf{W}}$ leads to

$$n\,\overline{\mathbf{D}}\,\overline{\mathbf{L}}\,\overline{\mathbf{D}} \preceq \frac{\theta}{2}\left(\overline{\mathbf{L}}\,\overline{\mathbf{D}} + \overline{\mathbf{D}}\,\overline{\mathbf{L}}\right). \tag{28}$$

Preconditioning for **homogeneous** problem $f_i(x) \equiv \frac{1}{2}x^\top \mathbf{L} x$

Define $\mathbf{D} = \mathrm{Diag}(\mathbf{L})$. Then, the original problem can be converted to

$$f_i(\mathbf{D}^{-\frac{1}{2}}x) = \frac{1}{2}x^\top \underbrace{\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}\right)}_{\tilde{\mathbf{L}}} x. \tag{29}$$

Preconditioning for **homogeneous** problem $f_i(x) \equiv \frac{1}{2}x^\top \mathbf{L} x$

Define $\mathbf{D} = \mathrm{Diag}(\mathbf{L})$. Then, the original problem can be converted to

$$f_i(\mathbf{D}^{-\frac{1}{2}} x) = \frac{1}{2} x^\top \underbrace{\left(\mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}\right)}_{\tilde{\mathbf{L}}} x. \tag{29}$$

Combined with `Perm-1` sketches

$$\mathbb{E}\left[\overline{\mathbf{B}}^k\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbf{C}_i \, \tilde{\mathbf{L}} \, \mathbf{C}_i\right] = n\mathrm{Diag}(\tilde{\mathbf{L}}) = n\mathbf{I}. \tag{30}$$

The resulting convergence guarantee is

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left[\left\|\nabla f(x^k)\right\|_{\mathbf{I}}^2\right] \leq \frac{2\lambda_{\max}(\tilde{\mathbf{L}})\left(f(x^0) - \mathbb{E}\left[f(x^K)\right]\right)}{K}. \tag{31}$$

Heterogeneous sketch preconditioning

Modification of `Perm-1`:

$$\tilde{\mathbf{C}}_i := \sqrt{n/\left[\mathbf{L}_i\right]_{\pi_i,\pi_i}}\, e_{\pi_i} e_{\pi_i}^\top. \tag{32}$$

In this case

$$\mathbb{E}\left[\tilde{\mathbf{C}}_i \mathbf{L}_i \tilde{\mathbf{C}}_i\right] = \mathbf{I} \quad \text{and} \quad \mathbb{E}\left[\overline{\mathbf{B}}^k\right] = \mathbf{I} \tag{33}$$

Convergence guarantee

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left[\left\|\nabla f(x^k)\right\|_{\mathbf{I}}^2\right] \leq \frac{2\lambda_{\max}(\overline{\mathbf{L}})\left(f(x^0) - \mathbb{E}\left[f(x^K)\right]\right)}{K}. \tag{34}$$