Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication : A Summary and Extension

September 29, 2025

1 General Context and Challenges

For economic and time reasons, we aim to reduce the resources required to train a model. To achieve this, we can act on various elements: - At a macro scale: logistics, how to connect elements, the design of the global model, etc. - At a micro scale: the choice of algorithm, data processing, number of iterations, etc.

Data exchange helps reduce constraints on one scale; collaboration enables parallel computation and accelerates processes. However, as collaboration increases, communication becomes a bottleneck.

Collaboration also has a security advantage through decentralization, by avoiding a single node aggregating all the data.

This requires the sharing of data in proportional quantities to $d \cdot weight_{\text{unit}}$, which motivates an in-depth study to reduce either factor without significantly degrading the final model.

Under certain assumptions, it is possible to maintain efficient convergence while reducing the required resources using compression, as proposed by approaches like CHOCO-Gossip and CHOCO-SGD.

2 State of the Art and Limitations of Current Approaches

Decentralization has been a topic of interest since the 2000s, initially with simple goals such as solving the average computation problem, and later for optimization purposes in works like those of Kempe et al. (2003) and Xiao and Boyd (2004). New proposals address more complex problems (decentralized optimization of a function), but they require strong assumptions (i.i.d distributions, strongly convex functions, etc.) and achieve at best sub-linear convergence.

Similarly, compression has been explored and includes techniques such as sparsification and quantization. Both aim to reduce the amount of transmitted data (by transmitting less, e.g., many zeros, or by encoding with fewer bits). However, existing algorithms require strong precision constraints.

The CHOCO-SGD method stands out by allowing arbitrary compression while ensuring linear convergence (not just local, as in some other methods).

This is achieved through several advancements in the algorithm's design:
1. It maintains a copy of the state it shares with others (what it has sent), quantifying and addressing the error introduced by compression to eliminate it.
2. A node shares a compressed update of the direction to correct the position that others perceive, rather than sharing an exact position. This avoids cumulative effects of relative compression errors.

Thus, along with certain other adjustments and correct assumptions, the paper demonstrates that linear convergence can be expected.

3 Average Consensus and Gossip Algorithm

Goal: Find the average of n points $(x_1, \ldots, x_n)$.

Each point represents a node of a graph G. Edges are defined by w_{ij}, and the gossip matrix is $W = (w_{ij})$. It is assumed that W is symmetric and doubly stochastic.

The idea of the gossip algorithm is to compute, for each worker i, a convex combination between its current state $x_i(t)$ and the states of its neighbors (including itself), weighted by w_{ij}:

$$x_i^{(t+1)} \leftarrow x_i^{(t)} + \gamma \sum_{j:\{i,j\}\in E} w_{ij}(x_j^{(t)} - x_i^{(t)}) = x_i^{(t)}(\gamma - 1) + \gamma \sum_{j:\{i,j\}\in E} w_{ij} x_j^{(t)}$$

4 Compression and Choco-Gossip

Compression

Compression in computer science is a technique used to reduce the amount of data transmitted or stored by removing redundancy. In the context of decentralized algorithms like the gossip algorithm, compression aims to reduce the communication burden between nodes while preserving essential information. For example:

- **Sparsification:** Transmitting only significant (non-zero) coordinates of a vector.
- **Quantization:** Reducing the precision of transmitted values by rounding them to a smaller number of bits.
- **Top-k compression:** Transmitting only the k largest absolute values of a vector while setting the rest to zero.

These methods allow for reduced data transmission between nodes, which lowers communication costs, but they introduce a loss of information that must be managed to ensure algorithm convergence.

Application of Compression

In the context of the gossip algorithm, the goal is to compute the average of data across nodes. Instead of directly transmitting a node's value x_i to its neighbors, we transmit a compressed version $Q(x_i)$, where Q is a compression operator. However, if we directly apply this modification to the gossip algorithm, the scheme no longer converges because compression disrupts the invariant of the algorithm (the preservation of the average).

For example, consider a vector $x_i = [1.2, -0.8, 0.3, 0.0]$:

- With **sparsification at a threshold**, $Q(x_i) = [1.2, 0, 0, 0]$ (only values greater than 0.5 in absolute magnitude are preserved).
- With **2-bit quantization**, $Q(x_i) = [1, -1, 0, 0]$ (each value is rounded to the nearest integer).
- With **Top-k compression**, where $k = 2$, $Q(x_i) = [1.2, -0.8, 0, 0]$ (only the two largest absolute values are preserved).

These techniques reduce the size of transmitted data but modify the original values. This requires adjustments in the algorithm to guarantee convergence.

Convergence Issue

If each node simply applies the compression function Q, the global average will no longer be preserved after each iteration, which can lead to divergence of the process. To address this issue, the *Choco-Gossip* algorithm introduces auxiliary variables $\hat{x}_i$ and applies compression only to the updates, as explained in the next section.

We now aim to compress the communication in the gossip algorithm. This means that each neighbor receives $Q(x_i)$ instead of x_i, where Q is a compression operator. However, if we simply apply the gossip algorithm with this modification, the scheme no longer converges because the average is not preserved between steps.

Algorithm

The article introduces the *Choco-Gossip* algorithm, which manages to converge for a wide range of compression operators. The main idea of this algorithm is to avoid applying the compression operator directly on the variables x_i. Instead, for each worker, a second variable $\hat{x}_i$ is introduced, and the compression is applied to $x_i^{(t+1)} - x_i^{(t)}$ at each iteration.

Algorithm 1 CHOCO-GOSSIP

Input: Initial values $x_i^{(0)} \in \mathbb{R}^d$ on each node $i \in [n]$,
stepsize γ, communication graph $G = ([n], E)$,
mixing matrix W, initialize $\hat{x}_i^{(0)} := 0$ for all i.
for $t = 0, \dots, T-1$ **do do in parallel for all workers** $i \in [n]$
  $x_i^{(t+1)} := x_i^{(t)} + \gamma \sum_{j:\{i,j\}\in E} w_{ij}(\hat{x}_j^{(t)} - \hat{x}_i^{(t)})$
  $q_i^{(t)} := Q(x_i^{(t+1)} - \hat{x}_i{}^{(t)})$
  for neighbors $j : \{i,j\} \in E$ (including $i \in E$) **do**
    Send $q_i^{(t)}$ and receive $q_j^{(t)}$
    $\hat{x}_j^{(t+1)} := \hat{x}_j^{(t)} + q_j^{(t)}$
  end for
end for

We provide a detailed explanation of the general algorithm in the next section (take $f_i(x) = \frac{1}{2}(x - x_i)^2$.)

5 CHOCO-SGD

The goal of the algorithm is to perform decentralized stochastic optimization. Thus, to optimize the average of the functions $f_i(x) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i} F_i(x, \xi_i)$ where F_i is the loss function of the node i and $\mathcal{D}_i$ the distribution of this same node. In the article $\mathcal{D}_i$ are supposed to be either iid or discrete with disjoint support. Here is the algorithm of interest we're going to comment:

Algorithm 2 CHOCO-SGD

Require: Initial values $x_i^{(0)} \in \mathbb{R}^d$ on each node $i \in [n]$, consensus stepsize γ, SGD stepsizes $\{\eta_t\}_{t\geq 0}$, communication graph $G = ([n], E)$, and mixing matrix W
1: Initialize $\hat{x}_i^{(0)} \leftarrow 0$ for all i
2: **for** $t = 0, \dots, T-1$ **in parallel** for all workers $i \in [n]$ **do**
3: Sample $\xi_i^{(t)}$, compute gradient $g_i^{(t)} \leftarrow \nabla F_i(x_i^{(t)}, \xi_i^{(t)})$
4: $x_i^{(t+1/2)} \leftarrow x_i^{(t)} - \eta_t g_i^{(t)}$
5: $x_i^{(t+1)} \leftarrow x_i^{(t+1/2)} + \gamma \sum_{j:\{i,j\}\in E} w_{ij}(\hat{x}_j^{(t)} - \hat{x}_i^{(t)})$
6: $q_i^{(t)} \leftarrow Q(x_i^{(t+1)} - \hat{x}_i^{(t)})$
7: **for** neighbors j: $\{i,j\} \in E$ (including $\{i\} \in E$) **do**
8: Send $q_i^{(t)}$ and receive $q_j^{(t)}$
9: $\hat{x}_j^{(t+1)} \leftarrow q_j^{(t)} + \hat{x}_j^{(t)}$
10: **end for**
11: **end for**

Explanation of CHOCO-SGD Steps

For all nodes i:

Step 1. Local Gradient Descent (Lines 3–4): Perform local gradient descent on node i:

$$g_i^{(t)} \leftarrow \nabla F_i(x_i^{(t)}, \xi_i^{(t)}), \quad x_i^{(t+1)} \leftarrow x_i^{(t)} - \eta_t g_i^{(t)}.$$

Step 2. Convex Combination with Neighbors (Line 5): Compute a convex combination between the local value x_i and the neighbors' values from the previous step ($\hat{x}_j$). If there is no compression and the consensus averaging scheme is applied, this step becomes:

$$\hat{x}_i^{(t+1)} \leftarrow x_i^{(t+1/2)}(1-\gamma) + \gamma \sum_{j:\{i,j\}\in E} w_{ij} x_j^{(t)}.$$

This corresponds to the gossip algorithm. A lower value of γ gives more importance to local gradient iterations.

Step 3. Compression and Communication (Lines 6–8): Compress the updates and communicate with neighbors:

$$q_i^{(t)} \leftarrow Q(x_i^{(t+1)} - \hat{x}_i^{(t)}),$$

where Q represents the compression operator. Each node sends $q_i^{(t)}$ and receives $q_j^{(t)}$ from its neighbors.

Step 4. Variable Update (Line 10): Update the variable $\hat{x}_j$. If there is no quantization, the update is:

$$\hat{x}_j^{(t+1)} = x_j^{(t+1)}.$$

Otherwise, the update is:

$$\hat{x}_j^{(t+1)} = Q(x_j^{(t+1)} - \hat{x}_j^{(t)}) + \hat{x}_j^{(t)} \approx x_j^{(t+1)},$$

where $\approx$ means that $\hat{x}_j^{(t+1)}$ approaches $x_j^{(t+1)}$ as the quantization parameter δ approaches 1.

6 Results

To evaluate the performance of the CHOCO-algorithm, two datasets are used:

- **Epsilon:** A dense, low-dimensional dataset.
- **RCV1:** A sparse, high-dimensional dataset.

The tests are performed on logistic regression (with penalization) where the m data samples are evenly distributed among the n workers in two settings:

1. **Random Setting:** Data points are randomly assigned to workers.
2. **Sorted Setting:** Each worker receives data samples from only one class.

The suboptimality is observed as:

$$f(\bar{x}^{(t)}) - f^*,$$

where $f(\bar{x}^{(t)})$ is the function value at iteration t and f^* is the optimal function value.

In the first case (random settings), topology doesn't have a significant impact on the convergence rate, at least when the number m of data is much greater than the number n of workers.

Topologies Considered

Two types of topologies are analyzed:

- **Fully-Connected Topology:** This is considered the easier case due to higher communication capability. In this case, there is no significant impact of the method used.
- **Ring Topology:** This is a harder scenario as communication is limited, resulting in slower convergence rates. Experiments show that the convergence rate is smaller in this case.

The comparisons are primarily focused on the ring topology under the sorted data setting, where workers with the same label form two connected clusters.

Performance of CHOCO-SGD

CHOCO-SGD demonstrates the following advantages:

- **Convergence:** CHOCO-SGD converges successfully in the ring topology setting, whereas baselines such as ECD and DCD struggle to converge.
- **Communication Efficiency:** In terms of convergence speed with respect to the number of transmitted bits CHOCO-SGD outperforms classic decentralized SGD with plain communication. The advantage is even more pronounced on the sparse dataset (RCV1).

7 Contribution

7.1 Implementation and visualization

To complement the theoretical contributions of this article, we have implemented the proposed algorithms and developed graphical visualizations to illustrate their behavior. These visualizations provide an intuitive understanding of the algorithms' dynamics and demonstrate their effectiveness in various scenarios (cf oral presentation). The implementation and visualizations were tested on a range of examples to validate their correctness and practical relevance. The source code for this project is available on GitHub.

7.2 Theoretical improvements

In this work, we have also achieved certain theoretical improvements, refining the analysis and extending the applicability of the proposed algorithms. These advancements are shown in the next parts.

Directed Graph

CHOCO-gossip

In this section, we show that the assumption on the symmetry of the gossip matrix can be removed. In fact, we explored this because, with a non-symmetric gossip matrix, the representation graph becomes directed, which can be very interesting as it allows for scenarios where some agents communicate in only one direction.

To prove the convergence of the choco-gossip algorithm, we now only suppose the gossip matrix W to be doubly stochastic (but not necessarily symmetric).

To prove the convergence, we rely on the proof from the article. However, *Lemma 16* does not apply anymore. Thus, we introduce this variation:

Lemma 1. *Let W be a doubly stochastic matrix with the second largest eigenvalue $1-\rho = |\lambda_2| < 1$. Then:*

$$\left\| W - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \right\|_2 \leq 1 - \rho.$$

Proof. See Appendix

□

Also, the symmetry assumption (used to diagonalize W) is used in Lemmas 17 and 18 when replacing $\|W - I\|_2$ by $\beta = \max_i(1-\lambda_i)$. If W is not diagonalizable, this equality does not hold anymore. However, we have the following inequality: $\|W - I\|_2 \leq 1 + |\lambda_{max}| = 2$. Thus, the inequality remains true with $\beta = \|W - I\|_2$. Because the only assumption we need on β is for it to be less than 2, the rest of the proof remains the same.

Finally, $\|\mathbf{I}+\gamma(\mathbf{I}-W)\|_2 = 1+\gamma\|\mathbf{I}-W\|_2 = 1+\gamma\beta$ becomes $\|\mathbf{I}+\gamma(\mathbf{I}-W)\|_2 \leq 1+\gamma\|\mathbf{I}-W\|_2 = 1+\gamma\beta$ with β as we defined above

The rest of the proof stay the same

CHOCO-SGD

The proof of the convergence of the Choco-SGD algorithm relies on the convergence of the Choco-Gossip algorithm, which we showed does not require the assumption of the symmetry of the gossip matrix W. (See Assumption 3: $\mathbb{E}_h \Psi(X^+, Y^+) \leq (1-p)\Psi(X, Y)$).

In the rest of the proof, W is not mentioned anymore (nor is its symmetry). Thus, Theorem 19 and the convergence of the Choco-SGD algorithm still hold with W asymmetric.

Random gossip matrix

We now want to consider the case where the graph is not fixed. We model this by setting W, the gossip matrix, as a random gossip matrix W_t that changes after each step of the gossip algorithm. The only assumption we need is $\mathbb{P}(\rho_t > 0) > 0$. Let's remark that if $\rho_t = 0$, the proof leads us to take $\gamma_t = 0$ i.e. not moving.

Our goal is to show that e_{t+1} goes to 0 (a.s.) as t goes to infinity.

The beginning of the proof is the same, and we obtain:

$$e_{t+1} \leq \max\{\eta_1(\gamma), \xi_1(\gamma)\} \cdot e_t.$$

By setting $\gamma_t = \gamma^*$, as defined in proof 2 (20), we obtain:

$$e_{t+1} \leq \lambda_t \cdot e_t,$$

with $\lambda_t = 1 - \rho_t^2 \cdot \delta/82$.

Thus,

$$\frac{e_t}{e_0} = \prod_{k=0}^{t} \lambda_k,$$

and $(\lambda_t)_{t>0}$ is i.i.d. with $\mathbb{E}[\lambda] = c < 1$ (because $\mathbb{P}(\rho_t > 0) > 0$).

By the strong law of large numbers (we have $\lambda_t > 0$):

$$\frac{1}{T} \sum_{k=1}^{T} \ln(\lambda_k) \to \mathbb{E}[\ln(\lambda)] < 0,$$

which implies that

$$\sum_{k=1}^{t} \ln(\lambda_k) \to -\infty,$$

and therefore,

$$\prod_{k=0}^{t} \lambda_k \to 0 \quad \text{a.s.}$$

$$e_t \to 0 \quad \text{a.s.}$$

Moreover, the convergence can be controlled in expectation:

$$\mathbb{E}(e_{t+1}) \leq \mathbb{E}(\lambda_t \cdot e_t) = \mathbb{E}(\lambda) \cdot \mathbb{E}(e_t).$$

Thus,

$$\mathbb{E}(e_t) \leq (\mathbb{E}(\lambda))^t \cdot \mathbb{E}(e_0),$$

which shows linear convergence in expectation.

Remark: The gossip algorithm presented here differs slightly from the original because the step size γ_t is now a variable that depends on W_t rather than being fixed. We also proved that we can keep a fixed step and linear convergence if the weights of the gossip matrix are taken among a finite family and the graph is always strongly connected. You can check the appendix for the proof.

Asynchronous choco-gossip

In this final section, we propose an algorithm based on the asynchronous Choco-Gossip algorithm. Each worker i has an internal Poisson clock and exchanges information with one of its neighbors with probability P_{ij} whenever its clock ticks. We have implemented this algorithm and tested it on simple examples to verify its functionality.

Algorithm 3 CHOCO-GOSSIP

Input: Initial values $x_i^{(0)} \in \mathbb{R}^d$ on each node $i \in [n]$,
stepsize γ, communication graph $G = ([n], E)$,
mixing matrix W, initialize $\hat{x}_i^{(0)} := 0$ for all i.
for each communication step between i and k **do**
  $x_i^{(t+1)} := x_i^{(t)} + 1/2(\hat{x}_k^{(t)} - \hat{x}_i^{(t)})$
  $q_i^{(t)} := Q(x_i^{(t+1)} - \hat{x}_i{}^{(t)})$
  for neighbors $j : \{i, j\} \in E$ (including $i \in E$) **do**
    Send $q_i^{(t)}$ and receive $q_j^{(t)}$
    $\hat{x}_j^{(t+1)} := \hat{x}_j^{(t)} + q_j^{(t)}$
  end for
  do the same for the worker k
end for

Appendix

Proof

Proof. lemma 1

Let $W = U\Lambda V^\top$ be the SVD decomposition of W. Because of the stochastic property of W, its first write and left singular vector are $u_1 = v_1 = \frac{1}{\sqrt{n}}\mathbf{1}$. Moreover U and V are orthogonal matrices. Thus :

$$U \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} V^\top = u_1 v_1^\top = \frac{1}{n}\mathbf{1}\mathbf{1}^\top.$$

Hence,

$$\left\| W - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \right\|_2 = \left\| U\Lambda V^\top - U \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} V^\top \right\|_2 = \left\| \Lambda - \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \right\|_2.$$

This simplifies to:

$$\left\| W - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \right\|_2 = 1 - \rho.$$

□

Random graph

If we suppose the graph to be randomly selected (cf part on random gossip matrix) among a finite number of possible graph, we can easily control the convergence of the gossip algorithm.

For all n, $\rho_n = 1 - |\lambda_2(W_n)| > 0$, and there exists $\rho_{\min} = \min_n(\rho_n) > 0$. Similarly, there exists a $\beta_{\max}$ such that $\beta_n \leq \beta_{\max}$ for all n.

We have defined (cf proof from the article):

$$\eta_1(\gamma) := (1 - \rho\gamma)^2(1 + \alpha_1) + (1 - \delta)\gamma^2\beta^2(1 + \alpha_2^{-1}),$$

and

$$\xi_1(\gamma) := \gamma^2\beta^2(1 + \alpha_1^{-1}) + (1 - \delta)(1 + \gamma\beta)^2(1 + \alpha_2).$$

Since $\rho_n \geq \rho_{\min}$ and $\beta_n \leq \beta_{\max}$, it follows that:

$$\eta_1(\gamma) \leq \eta_{\beta_{\max},\rho_{\min}}(\gamma) := (1 - \rho_{\min}\gamma)^2(1 + \alpha_1) + (1 - \delta)\gamma^2\beta_{\max}^2(1 + \alpha_2^{-1}),$$

and

$$\xi_1(\gamma) \leq \xi_{\beta_{\max},\rho_{\min}}(\gamma) := \gamma^2\beta_{\max}^2(1 + \alpha_1^{-1}) + (1 - \delta)(1 + \gamma\beta_{\max})^2(1 + \alpha_2).$$

By choosing the parameters as follows:

$$\alpha_1 = \frac{\gamma\rho_{\min}}{2}, \quad \alpha_2 = \frac{\delta}{2},$$

and with

$$\gamma^\star := \frac{\rho_{\min}\delta}{16\rho_{\min} + \rho_{\min}^2 + 4\beta_{\max}^2 + 2\rho_{\min}\beta_{\max}^2 - 8\rho_{\min}\delta},$$

we can bound the terms as follows:

$$\eta_1(\gamma^\star) < \eta_{\beta_{\max},\rho_{\min}}(\gamma^\star) < 1 \quad \text{and} \quad \xi_1(\gamma^\star) < \xi_{\beta_{\max},\rho_{\min}}(\gamma^\star) < 1.$$

Thus, the assumption can be weakened since we can find a step size γ for a fixed δ that ensures linear convergence.