Review of Research Paper number 3 - EXTRA Algorithm

Gaspard Blaise, Dimitri De Saint Guilhem, Shun Ye Chen

version: April 2, 2025

1 Introduction

1.1 Summary of the paper

This paper presents **EXTRA**, a decentralized optimization algorithm for multi-agent networks, where each agent solves a private convex optimization problem while only communicating with its neighbors. The main challenge in decentralized optimization is achieving **consensus** and **convergence** efficiently. Unlike **Decentralized Gradient Descent (DGD)**, which requires diminishing step sizes to ensure exact convergence, EXTRA allows **fixed large step sizes**, leading to **faster and more stable convergence**. The paper proves that EXTRA achieves an **ergodic convergence rate of $O(1/k)$ for general convex problems and linear convergence when the global function is strongly convex**. The algorithm is developed by modifying DGD using gradients from the last two iterations, avoiding DGD's slow or inexact convergence issues. It improves upon standard **first-order decentralized methods**, including DGD, subgradient methods, and decentralized dual averaging. Applications include distributed machine learning, sensor networks, and multi-agent control.

1.2 Problem Formulation

1.2.1 Problem Statement

This paper focuses on *decentralized consensus optimization*, a problem defined on a connected network and solved cooperatively by n agents:

$$\min_{x\in\mathbb{R}^p} \bar{f}(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where each agent i has its own objective function $f_i : \mathbb{R}^p \to \mathbb{R}$, which is convex and privately known by the agent.

We assume that the functions f_i are continuously differentiable. The paper introduces a novel first-order algorithm to solve this problem in a decentralized manner.

In this study, we focus on the synchronous case, meaning that all agents perform their iterations at the same time intervals.

1.2.2 Notations

Each agent i holds a *local copy* of the global variable x, denoted as $x_{(i)} \in \mathbb{R}^p$, with its value at iteration k given by $x^k_{(i)}$. The aggregate objective function is defined as:

$$\mathbf{f}(\mathbf{x}) \triangleq \sum_{i=1}^{n} f_i(x_{(i)}), \tag{2}$$

where the stacked variable representation is:

$$\mathbf{x} \triangleq \begin{pmatrix} x_{(1)}^\top \\ x_{(2)}^\top \\ \vdots \\ x_{(n)}^\top \end{pmatrix} \in \mathbb{R}^{n\times p}. \tag{3}$$

The gradient of $\mathbf{f}(\mathbf{x})$ is given by:

$$\nabla \mathbf{f}(\mathbf{x}) \triangleq \begin{pmatrix} \nabla^\top f_1(x_{(1)}) \\ \nabla^\top f_2(x_{(2)}) \\ \vdots \\ \nabla^\top f_n(x_{(n)}) \end{pmatrix} \in \mathbb{R}^{n \times p}. \tag{4}$$

Each row of $\mathbf{x}$ and $\nabla \mathbf{f}(\mathbf{x})$ corresponds to an agent. The vector $\mathbf{x}$ is said to be *consensual* if all its rows are identical, i.e., $x_{(1)} = \cdots = x_{(n)}$. The analysis holds for all $p \geq 1$, with $p = 1$ simplifying the notation.

1.3 Assumptions

We consider the mixing matrices $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ and $\tilde{W} = [\tilde{w}_{ij}] \in \mathbb{R}^{n \times n}$ used in the EXTRA algorithm. These matrices satisfy the following assumptions:

1. **Decentralized Property**: If agents i and j are not connected in the network, i.e., $(i, j) \notin \mathcal{E}$, then
$$w_{ij} = \tilde{w}_{ij} = 0.$$
2. **Symmetry**: The mixing matrices are symmetric:
$$W = W^\top, \quad \tilde{W} = \tilde{W}^\top.$$
3. **Null Space Property**: The null spaces satisfy:
$$\text{null}\{W - \tilde{W}\} = \text{span}\{1\}, \quad \text{null}\{I - \tilde{W}\} \supseteq \text{span}\{1\}.$$
4. **Spectral Property**: The matrix $\tilde{W}$ is positive definite, and the following inequality holds:
$$\frac{I + W}{2} \succeq \tilde{W} \succeq W.$$

::

2 Result: theory and practice

2.1 Theory

Describe the main theorems of the paper and the results therein in mathematical terms, **by using the notation of the lecture notes.** Interpret the results [1 page max]

2.1.1 Standard Decentralized Gradient Descent (DGD)

A common approach to solving (P) in a decentralized setting is the Decentralized Gradient Descent (DGD) algorithm:

$$x_i^{k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_j^k - \alpha_k \nabla f_i(x_i^k), \tag{5}$$

where w_{ij} are entries of a doubly stochastic mixing matrix W, and α_k is a step size. However, DGD suffers from consensus mismatches and requires a diminishing step size to ensure exact convergence.

2.1.2 The EXTRA Algorithm

The paper introduces the **EXTRA (Exact First-Order Algorithm for Decentralized Consensus Optimization)** algorithm, which improves upon DGD by allowing a fixed step size while ensuring exact convergence. The update rule for EXTRA is given by:

$$x^{k+2} = (I + W)x^{k+1} - \tilde{W}x^k - \alpha \left(\nabla f(x^{k+1}) - \nabla f(x^k) \right), \tag{6}$$

where:

- $W \in \mathbb{R}^{n\times n}$ is a symmetric, doubly stochastic mixing matrix ensuring consensus,
- $\tilde{W} \in \mathbb{R}^{n\times n}$ is another mixing matrix, often chosen as $\tilde{W} = (W + I)/2$,
- $\alpha > 0$ is the step size,
- $\nabla f(x^k)$ represents the gradient of the local objective function at iteration k.

Breaking Down the Update Step for Each Agent At each agent i, the decentralized computation is performed as follows:

1. **Step 1 (Initialization Update):**

$$x_i^1 = \sum_{j=1}^{n} w_{ij} x_j^0 - \alpha \nabla f_i(x_i^0), \quad i = 1, \ldots, n. \tag{7}$$

2. **Step 2 (Iterative Update for $k \geq 0$):**

$$x_i^{k+2} = x_i^{k+1} + \sum_{j=1}^{n} w_{ij} x_j^{k+1} - \sum_{j=1}^{n} \widetilde{w}_{ij} x_j^k - \alpha \left(\nabla f_i(x_i^{k+1}) - \nabla f_i(x_i^k) \right), \tag{8}$$

   where each agent computes its local gradient $\nabla f_i(x_i^k)$ and communicates with its neighbors according to the mixing matrices.

Unlike **DGD**, which requires a vanishing step size for exact convergence, **EXTRA** achieves consensus with a fixed step size, leading to faster and more stable convergence.

2.1.3 EXTRA as Corrected DGD

The EXTRA algorithm can be interpreted as a Decentralized Gradient Descent (DGD) method with a cumulative correction term. The update rule for EXTRA is given by:

$$x^{k+1} = Wx^k - \alpha \nabla f(x^k) + \sum_{t=0}^{k-1} (W - \tilde{W}) x^t. \tag{9}$$

This cumulative correction term, $\sum_{t=0}^{k-1}(W-\tilde{W})x^t$, is crucial because it neutralizes the persistent gradient term $-\alpha\nabla f(x^k)$ in the subspace orthogonal to the consensus subspace span1. Without this correction, using a fixed step size $\alpha > 0$ would prevent consensus due to non-vanishing gradient terms. Thus, this correction ensures consensus and optimality conditions are met simultaneously, facilitating the linear convergence property observed with EXTRA.

2.1.4 Comparison with Existing Methods

- Unlike **DGD**, which requires diminishing step sizes for exact convergence, **EXTRA** allows a fixed step size.
- Compared to **Gradient Tracking**, EXTRA achieves exact convergence while maintaining a lower communication and computation cost.

Thus, EXTRA provides a significant improvement over existing decentralized optimization methods by achieving exact consensus with faster and more stable convergence.

2.2 Practice

We focus on applying the decentralized EXTRA algorithm for Kernel Ridge Regression. We first define a kernel ridge regression problem using the Euclidean kernel given by:

$$k(x, x_i) = \exp(-|x - x_i|^2). \tag{10}$$

We define the kernel matrix $K = [k(x_i, x_j)]i, j = 1, ..., n$. To reduce the computational complexity, we employ a Nystrom approximation by uniformly selecting at random a subset $\mathcal{M}$ of $m = \sqrt{n}$ points from the original n points. Thus, the function approximation becomes:

$$f(x) \approx \sum_{j \in \mathcal{M}} \alpha_j k(x, x_j). \tag{11}$$

The optimal parameter vector $\alpha \in \mathbb{R}^m$ is obtained by solving the following ridge regression-like optimization problem:

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^m} \frac{\sigma^2}{2} \alpha^T K_{nm} \alpha + \frac{1}{2} |y - K_{nm}\alpha|_2^2, \tag{12}$$

where $K_{nm} = [k(x_i, x_j)]_{i=1,...,n; j \in \mathcal{M}}$ and y is the stacked vector of observed outputs. To ensure strong convexity and simplify computations, we introduce an additional regularization term controlled by $\nu = 1$:

$$\frac{\nu}{2} |\alpha|_2^2, \tag{13}$$

yielding a smooth, strongly convex optimization problem.

Choice of W
We consider the Laplacian-based constant edge weight matrix defined as follows:

$$W = I - \frac{L}{\tau}, \tag{14}$$

where L is the Laplacian matrix of the graph and $\tau > \frac{1}{2}\lambda_{\max}(L)$ is a scaling parameter. When $\lambda_{\max}(L)$ is not readily available, a practical choice for τ is:

$$\tau = \max_{i \in \mathcal{V}} \{\deg(i)\} + \epsilon, \quad \text{with } \epsilon = 1. \tag{15}$$

This choice of weight matrix W ensures good diffusion of information across the network and is commonly used in consensus algorithms.

2.3 Results

We chose to compare EXTRA to DGD as it is an improvement of the latter. We used 2 versions of DGD, one with a constant step size and one with a diminishing one. We chose $s = 0.002$ as a constant step-size for both DGD (constant) and EXTRA, and

$$s(k) = \frac{0.002}{0.001\, k + 1}$$

for the diminishing one.

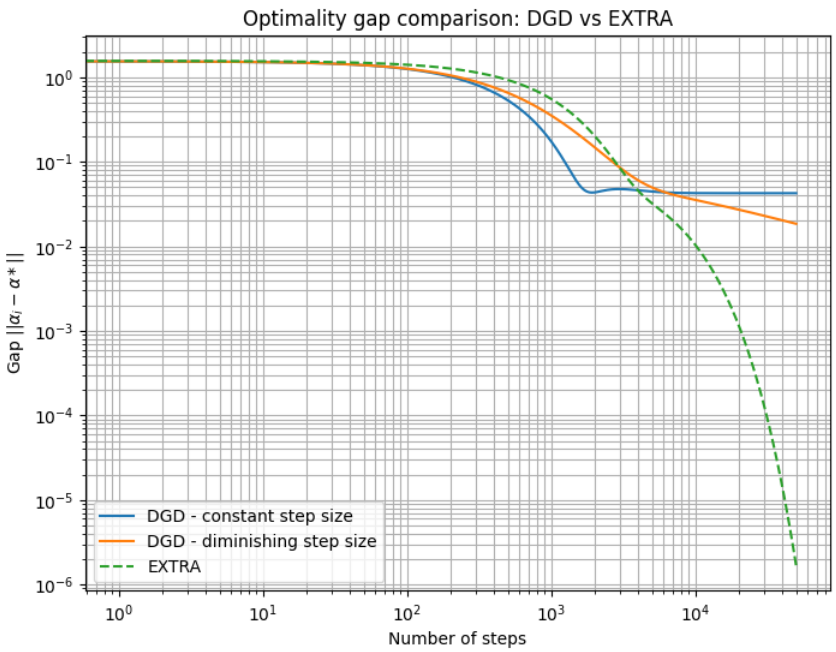

Figure 1: Optimality gap comparison between EXTRA and classical DGD methods.

Figure 1 illustrates the convergence behavior of EXTRA compared to standard DGD with constant and diminishing step sizes. We observe that EXTRA achieves significantly faster convergence with a consistently decreasing optimality gap, outperforming both constant and diminishing step-size DGD approaches.

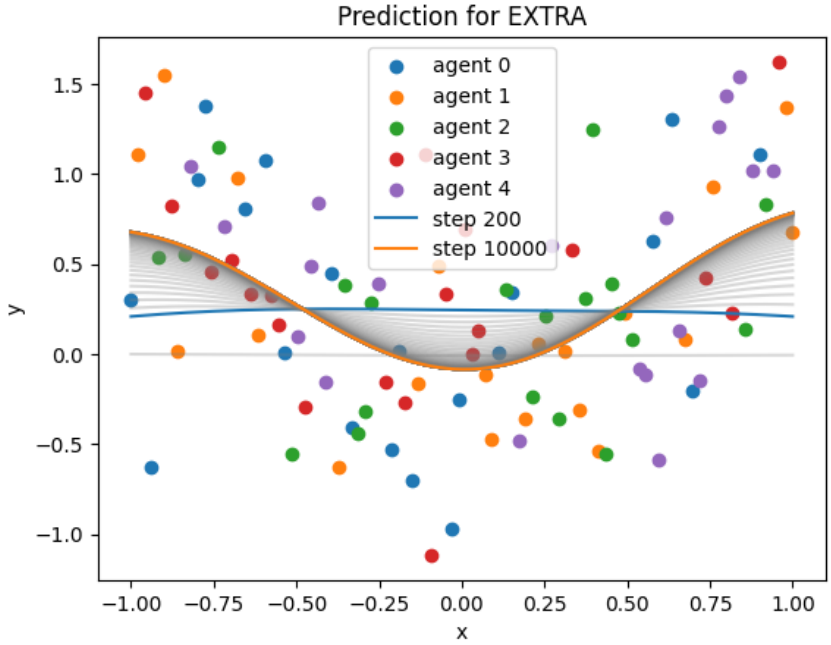

Figure 2: Reconstruction of EXTRA

Figure 2 illustrates the reconstruction results of the decentralized EXTRA algorithm at different iteration steps. Scattered points represent observations from 5 distinct agents, each color-coded. Solid lines represent the predictions at iteration steps 200 and 10,000, illustrating the algorithm's progression from initial stages (blue curve, step 200) towards convergence to an optimal consensus solution (orange curve, step 10,000).

3 Conclusion

The EXTRA algorithm provides an efficient and exact first-order method for decentralized consensus optimization. Unlike traditional decentralized gradient descent (DGD), it achieves exact convergence with a fixed step size, making it independent of network size and topology. EXTRA improves convergence rates by utilizing gradient corrections, ensuring linear convergence under strong convexity and an ergodic $O(1/k)$ rate for general convex functions. Its decentralized nature makes it well-suited for large-scale multi-agent systems and distributed learning applications.