DECENTRALIZED STOCHASTIC COMPOSITIONAL GRADIENT DESCENT FOR AUPRC MAXIMIZATION

Hongchang Gao,
Yubin Duan,
Yihan Zhan,
Jie Wu

Department of Computer and Information Sciences, Temple University

BACKGROUND

The Reality of Imbalanced Data:

- Real-world classification tasks are inherently skewed.
- Example: In online advertising Click-Through Rate (CTR) prediction, positive clicks are drastically outnumbered by non-clicks.

The Scale of the Problem:

- Modern datasets are not only massive but naturally distributed across multiple devices or data silos.

The Critical Need:

- We urgently need **distributed** optimization algorithms capable of handling large-scale, **imbalanced data classification**.

THE AUC SOLUTION

The "Accuracy Paradox":

- Standard loss functions fail on skewed data.
- A model predicting "all negative" on a 99% imbalanced dataset achieves 99% accuracy but is practically useless.

The Solution:

- Directly optimize the Area Under the Curve (AUC) metric.

AUROC (Receiver Operating Characteristic):

- Evaluates the true positive rate vs. false positive rate.

AUPRC (Precision–Recall Curve):

- Evaluates precision vs. recall. Highly sensitive to the minority (positive) class.
- In this paper, we will focus on the AUPRC maximization problem.

RELATED WORKS & CHALLENGE

Advances in AUPRC Maximization

The Paradigm Shift:

- Recent literature reformulated AUPRC maximization from an **intractable sorting problem** into a **stochastic compositional optimization problem**.

State–of–the–Art Solvers:

- Advanced methods like Stochastic Compositional Gradient Descent with Momentum (SCGDM) have been deployed to solve this.

The Bottleneck:

- All existing AUPRC optimization frameworks strictly assume a **single–machine setting**, making them incapable of handling distributed, large–scale data.

RELATED WORKS & CHALLENGE

Progress in Decentralized Optimization

The Baseline:

- Decentralized Stochastic Gradient Descent (DSGD) and communication–efficient variants have become the standard for training large–scale machine learning models across multiple devices.

The Bottleneck:

- Standard DSGD methods are designed for standard objectives, not the complex stochastic compositional optimization required by AUPRC.

RELATED WORKS & CHALLENGE

DSCGD & Its Critical Limitations

Recently, the first Decentralized Stochastic Compositional Gradient Descent (DSCGD) algorithm was introduced.

While a step forward, it is fundamentally inadequate for large-scale AUPRC tasks due to three critical issues:

- Impractical Batch Size
- Suboptimal Sample Complexity
- Structural Incompatibility: AUPRC requires estimating the rank of each specific sample.

PRELIMINARIES

Dataset:

- We are given n samples denoted as $\{(a_i, b_i)\}_{i=1}^{n}$, where features $a_i \in \mathbb{R}^d$ and labels $b_i \in \{-1, 1\}$.

Data Partition::

- Positive Set ($\mathfrak{D}_+$): Contains m positive samples.
- Negative Set ($\mathfrak{D}_-$): Contains n–m negative samples.

Imbalance Assumption:

- It is an imbalanced classification problem (typically $m \ll n - m$).

Model: A classifier $h(x; a_i)$ parameterized by weights x.

APPROXIMATING AVERAGE PRECISION (AP)

The AP Metric:

- AP is an unbiased estimator of AUPRC. The objective is to maximize AP:
- $AP = \frac{1}{m} \sum_{a_i \in \mathbb{D}_+} \frac{\sum_{j=1}^{n} l(x; a_j, a_i) \cdot \mathbb{I}(b_j = 1)}{\sum_{j=1}^{n} l(x; a_j, a_i)}$

Smooth Surrogate Loss:

- Standard indicator functions $\mathbb{I}$ are non-differentiable.
- We approximate $\mathbb{I}\left(\mathrm{h}(x; a_j) \geq \mathrm{h}(x; a_i)\right)$ using a smooth Sigmoid function:
- $l(x; a_i, a_j) = \frac{\exp\left(\rho \mathrm{h}(x; a_j) - \mathrm{h}(x; a_i)\right)}{1 + \exp\left(\rho \mathrm{h}(x; a_j) - \mathrm{h}(x; a_i)\right)}$

OPTIMIZATION TRANSFORMATION

The Vector Representation:

- To enable gradient–based optimization, the numerator and denominator components are packed into a 2D vector ω:
- $\omega(x; a_j, a_i) = \begin{bmatrix} [\mathbb{I}(b_j = 1) I(x; a_j, a_i)] \\ I(x; a_j, a_i) \end{bmatrix}$

The Compositional Objective:

- Maximizing AP boils down to a stochastic compositional minimization problem:
- $\min_x F(x) \triangleq \frac{1}{m} \sum_{i \in \mathbb{D}^+} f(g_i(x))$
- Inner–level function: $g_i(x) = \sum_{j=1}^{n} \omega(x; a_j, a_i) \in \mathbb{R}^2$ (accumulates the ranks).
- Outer–level function: $f(y) = -\frac{y_1}{y_2} \in \mathbb{R}$ (calculates the negative precision).

DECENTRALIZED SETUP

Network Topology:

- The system consists of K devices communicating over a graph $\mathcal{G} = \{V, W\}$.
- $V = \{v_k\}_{k=1}^{K}$ is the set of devices.
- $W = [w_{ij}] \in \mathbb{R}^{\mathbb{K}\times\mathbb{K}}$ is the adjacency matrix ($w_{ij} > 0$ if devices i and j are connected).

Local Data:

- Each device k holds its own local dataset $\mathcal{D}^k = \{\mathcal{D}_+^k, \mathcal{D}_-^k\}$.

Global Decentralized Objective:

- $\min_x F(x) \triangleq \frac{1}{K}\sum_{k=1}^{K} F^k(x) = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{m}\sum_{i\in\mathcal{D}_i^k} f\left(g_i^k(x)\right)\right)$

PROPOSED METHODS

The Goal:

- Develop a decentralized optimization framework specifically tailored for the AUPRC maximization problem.

The paper proposes two novel algorithms:

- SE-DSCGDM: Focuses on optimizing Sample Efficiency.
- CE-DSCGDM: Upgrades SE-DSCGDM to optimize Communication Efficiency.

PROPOSED METHODS

Stochastic Inner–Level Estimation

- Computing the exact inner–level function is too expensive.
- We need an unbiased stochastic approximation.

For device k at iteration t, the stochastic inner–level function value $\widetilde{g_i^k}$ and gradient $\nabla\widetilde{g_i^k}$ for the i–th positive sample are computed as:

- $\widetilde{g_i^k}(x_t^k) = \frac{n}{|\mathbb{S}_t^k|}\sum_{j\in\mathbb{S}_t^k}\omega^k\left(x_t^k; a_j^k, a_i^k\right)$
- $\nabla\widetilde{g_i^k}(x_t^k) = \frac{n}{|\mathbb{S}_t^k|}\sum_{j\in\mathbb{S}_t^k}\nabla\omega^k\left(x_t^k; a_j^k, a_i^k\right)$

- n: Total number of samples.
- $\mathbb{S}_t^k$: A randomly sampled minibatch from all data on device k.
- $|\mathbb{S}_t^k|$: The size of this minibatch.

PROPOSED METHODS

The Sample–Wise Moving Average Estimator

We introduce a sample–wise estimator (U) to track the historical inner–level function value for each specific positive sample independently:

$$\mathbf{U}_{i,t+1}^{k} = \begin{cases} (1-\beta)\,\mathbf{U}_{i,t}^{k} + \beta \dfrac{m}{|\mathcal{B}_t^k|}\tilde{g}_i^k(\mathbf{x}_t^k), & i \in \mathcal{B}_t^k \\ (1-\beta)\,\mathbf{U}_{i,t}^{k}, & \text{o.w.} \end{cases}$$

- $\mathcal{B}_t^k$: A minibatch of positive samples selected at iteration t.
- $\beta \in (0,1)$: The momentum/decay weight.
- o.w.: If the sample i is not in the current minibatch.

ALGORITHM 1: SE–DSCGDM

Core Innovation:

- Introduces a sample–wise moving average estimator ($U^k_{i,t+1}$) to track the inner–level function

Workflow:

- Each device computes its stochastic compositional gradient and updates local momentum.
- Devices update local models and communicate with neighbors at every single iteration.

Algorithm 1 SE-DSCGDM

Input: $\mathbf{x}_0^k = \mathbf{x}_0$, $\beta \in (0,1)$, $\gamma \in (0,1)$, $\eta > 0$.

1: **for** $t = 0, \cdots, T-1$, each device k **do**

2: Select a minibatch of samples $\mathcal{B}_t^k$ from $\mathcal{D}_+^k$ and a minibatch of samples $\mathcal{S}_t^k$ from $\mathcal{D}^k$ to compute:

3: $$\mathbf{U}_{i,t+1}^k = \begin{cases} (1-\beta)\,\mathbf{U}_{i,t}^k + \beta \frac{m}{|\mathcal{B}_t^k|} \tilde{g}_i^k\left(\mathbf{x}_t^k\right) & i \in \mathcal{B}_t^k \\ (1-\beta)\,\mathbf{U}_{i,t}^k & \text{o.w.} \end{cases}$$

4: $\mathbf{v}_t^k = \frac{1}{|\mathcal{B}_t^k|} \sum_{i \in \mathcal{B}_t^k} \nabla \tilde{g}_i^k(\mathbf{x}_t^k)^T \nabla f(\mathbf{U}_{i,t+1}^k)$

5: $\mathbf{m}_{t+1}^k = (1-\gamma)\mathbf{m}_t^k + \gamma \mathbf{v}_t^k$

6: $\tilde{\mathbf{x}}_{t+1}^k = \mathbf{x}_t^k - \eta \mathbf{m}_{t+1}^k$

7: $\mathbf{x}_{t+1}^k = \sum_{j \in \mathcal{N}_{v_k}} w_{kj} \tilde{\mathbf{x}}_{t+1}^j$

8: **end for**

CONVERGENCE OF SE–DSCGDM

The Convergence Formula:

- By setting appropriate learning rates and parameters, the convergence rate to an ϵ–accuracy stationary point is bounded by:
- $\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla F(\bar{x}_t)\right\|^2 \leq \Theta\left(\frac{1}{\sqrt{T}}\right) + \Theta\left(\frac{1}{(1-\lambda)^2 T}\right)$
- The first term $\Theta\left(\frac{1}{\sqrt{T}}\right)$ is the dominant optimization rate.
- The second term incorporates λ (the spectral gap of the network graph), representing the consensus error among devices.

Sample Complexity:

- Improved to $\Theta(1/\epsilon^4)$, matching the best single–machine methods.

Communication Complexity:

- $\Theta(1/\epsilon^4)$.

ALGORITHM 2: CE–DSCGDM

Periodic Communication.

- Instead of syncing every step, devices perform local momentum updates and only aggregate weights with neighbors every p iterations.
- If $mod(t+1,p)=0$, devices communicate. Otherwise, they only update locally.

Algorithm 2 CE-DSCGDM

Input: $\mathbf{x}_0^k = \mathbf{x}_0$, $\beta \in (0,1)$, $\gamma \in (0,1)$, $\eta > 0$, $p > 1$.

1: **for** $t = 0, \cdots, T-1$, each device k **do**

2: Select a minibatch of samples $\mathcal{B}_t^k$ from $\mathcal{D}_+^k$ and a minibatch of samples $\mathcal{S}_t^k$ from $\mathcal{D}^k$ to compute:

3: $$\mathbf{U}_{i,t+1}^k = \begin{cases} (1-\beta)\,\mathbf{U}_{i,t}^k + \beta \frac{m}{|\mathcal{B}_t^k|} \tilde{g}_i^k\left(\mathbf{x}_t^k\right), & i \in \mathcal{B}_t^k \\ (1-\beta)\,\mathbf{U}_{i,t}^k, & \text{o.w.} \end{cases}$$

4: $$\mathbf{v}_t^k = \frac{1}{|\mathcal{B}_t^k|} \sum_{i \in \mathcal{B}_t^k} \nabla \tilde{g}_i^k(\mathbf{x}_t^k)^T \nabla f(\mathbf{U}_{i,t+1}^k)$$

5: $$\mathbf{m}_{t+1}^k = (1-\gamma)\mathbf{m}_t^k + \gamma \mathbf{v}_t^k$$

6: $$\tilde{\mathbf{x}}_{t+1}^k = \mathbf{x}_t^k - \eta \mathbf{m}_{t+1}^k$$

7: $$\mathbf{x}_{t+1}^k = \begin{cases} \sum_{j \in \mathcal{N}_k} w_{kj} \tilde{\mathbf{x}}_{t+1}^j, & \text{mod(t+1,p)} = 0 \\ \tilde{\mathbf{x}}_{t+1}^k, & \text{o.w.} \end{cases}$$

8: **end for**

CONVERGENCE OF CE–DSCGDM

The Convergence Formula:

- With periodic communication $p = \Theta(T^{\alpha})$ where $\alpha \in (0,1/4)$, the new convergence bound is:
- $\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla F(\bar{x}_t)\right\|^2 \leq \Theta\left(\frac{1}{\sqrt{T}}\right) + \Theta\left(\frac{1}{(1-\lambda)^2 T^{1-2\alpha}}\right)$
- as long as $\alpha < 1/4$, the first term $\Theta\left(\frac{1}{\sqrt{T}}\right)$ still dominates.
- This means periodic communication does not degrade the overall convergence rate.

Sample Complexity:

- Retains the optimal $\Theta(1/\epsilon^4)$.

Communication Complexity:

- Dramatically improved to $\Theta(1/\epsilon^{4-4\alpha})$.

EXPERIMENTAL SETUP

Datasets:

- CIFAR10, CIFAR100, STL10, and CATvsDOG.

Imbalance injection:

- Converted to binary tasks, then randomly removed 98% of positive samples to create highly imbalanced datasets.

Hardware & Framework:

- HPC server with 8x NVIDIA Tesla Volta V100 GPUs and 512GB RAM.
- Implemented using PyTorch and OpenMPI backends for distributed training.

Model & Baselines:

- Classifier: ResNet 18.
- Baselines: SOAP and MOAP-V2 (parallelized via all-reduce for fair comparison).

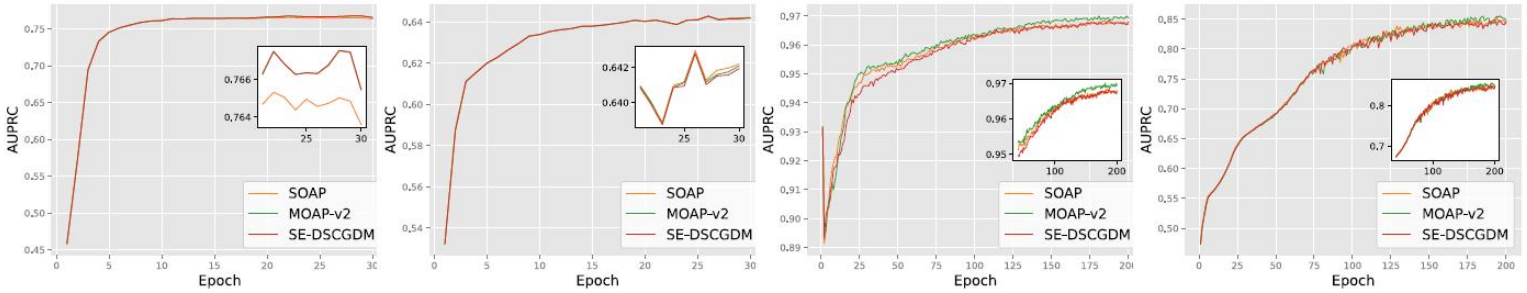
CONVERGENCE PERFORMANCE

Settings:

- Evaluated using K=4 devices with a Ring communication topology.

The Results:

- Across all four datasets, SE–DSCGDM successfully converges to almost the exact same testing AUPRC score as the centralized baselines (SOAP and MOAP–V2).
- Decentralizing the training process does not sacrifice the model's predictive performance.



(a) CIFAR10 (b) CIFAR100 (c) STL10 (d) CATvsDOG

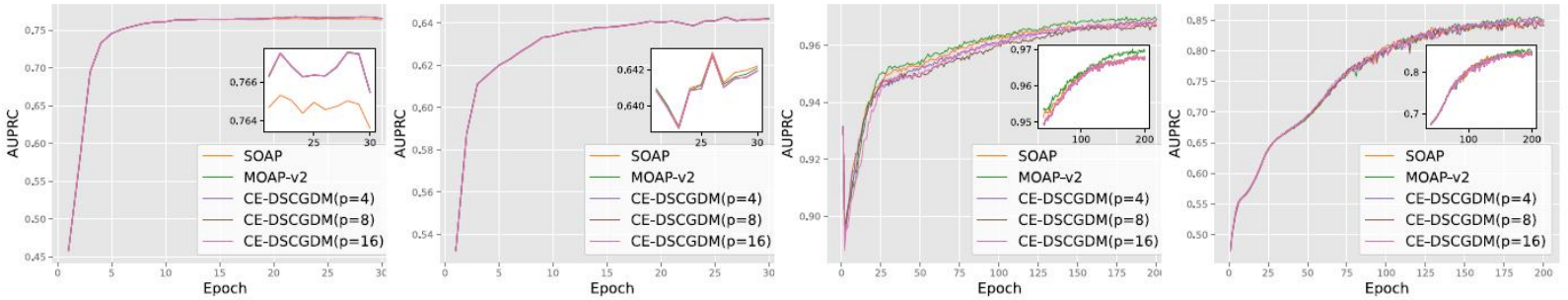
PERIODIC COMMUNICATION

Settings:

- Tested CE–DSCGDM with different communication periods ($p = 4, 8, 16$).

The Results:

- CE–DSCGDM with varying p values converges to the same final AUPRC score as the step–by–step SE–DSCGDM method.
- The periodic communication strategy is safe and does not impair convergence accuracy.



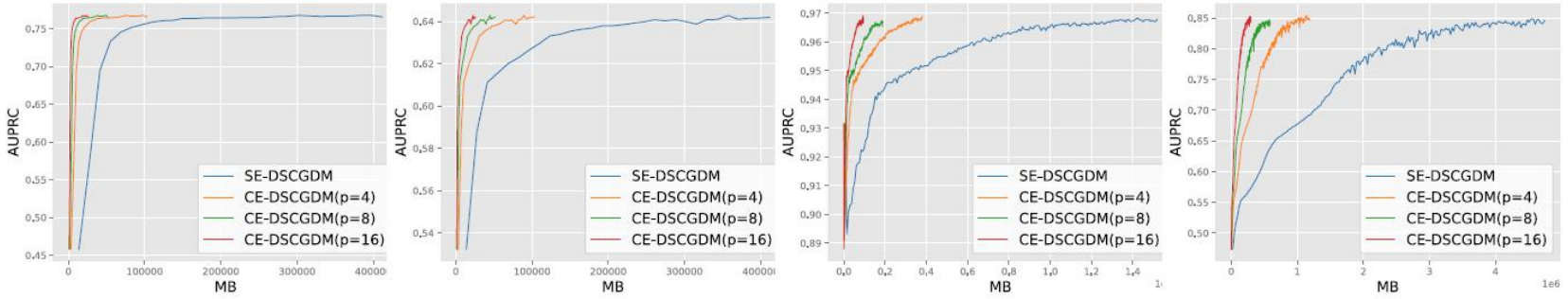
(e) CIFAR10 (f) CIFAR100 (g) STL10 (h) CATvsDOG

COMMUNICATION EFFICIENCY

Measure the direct trade–off between target AUPRC and the communication data cost (in MB).

The Results:

- All variants still achieve the same ultimate AUPRC score.
- CE–DSCGDM successfully slashes network bandwidth requirements, proving its communication efficiency.



(a) CIFAR10 (b) CIFAR100 (c) STL10 (d) CATvsDOG

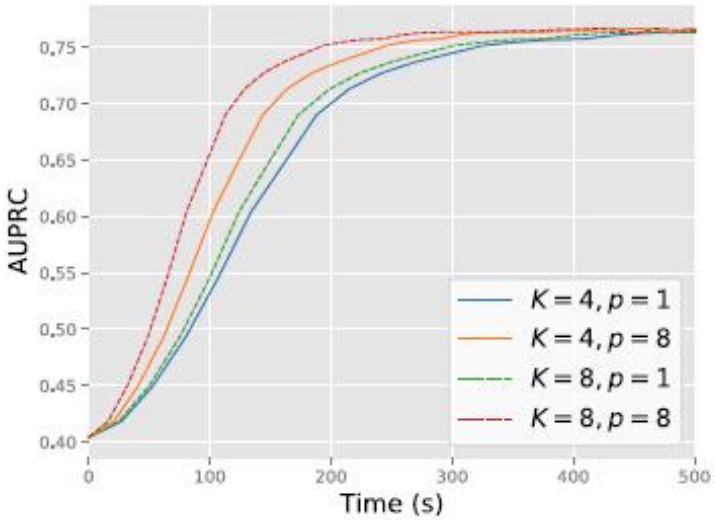
DISTRIBUTED SPEEDUP

Settings:

- Compared training time on CIFAR10 using K=4 vs. K=8 devices.

The Results:

- Both algorithms converge faster when utilizing more devices.
- The proposed framework scales efficiently, translating additional hardware directly into faster training times.



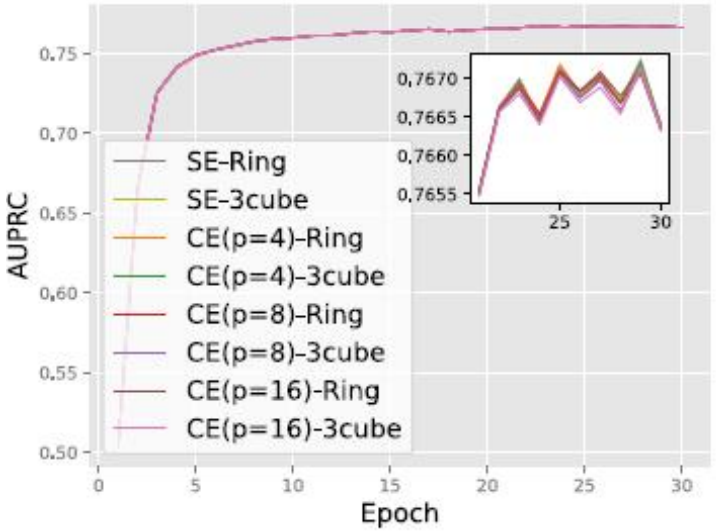
NETWORK TOPOLOGY ROBUSTNESS

Settings:

- Deployed on 8 devices, comparing a simple Ring graph vs. a highly connected 3–cube graph.

The Results:

- Both SE–DSCGDM and CE–DSCGDM achieve nearly identical testing AUPRC.
- The spectral gap of the network graph does not hinder convergence performance.



CONCLUSION

The Milestone

- Proposed the first decentralized optimization framework strictly tailored for large-scale AUPRC maximization.

SE-DSCGDM

- Achieved the optimal single-machine sample complexity of $\Theta(1/\epsilon^4)$ in a decentralized setting.
- Slashed the required batch size down to $\Theta(1)$.

CE-DSCGDM

- Introduced periodic communication to shatter the network bottleneck.
- Achieved unprecedented communication complexity of $\Theta(1/\epsilon^{4-4\alpha})$ without sacrificing sample efficiency.

Empirical Success

- Demonstrated robust convergence, communication efficiency, and linear distributed speedup across real-world imbalanced datasets.