Advanced Topics and Open Problems
Lecture 6: Advanced Topics and Open Problems

Stefan Vlaski[†] and Ali H. Sayed[⋆]

[†]Department of Electrical and Electronic Engineering, Imperial College London, UK
[⋆]Adaptive Systems Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland

IEEE ICASSP 2024 Short Course

Imperial College London

EPFL

Lecture Aims

- In the last lecture we derived a range of decentralized learning algorithms, which we classified as penalty-based, primal-dual and gradient-tracking based.
- We also derived incremental variants.
- In this lecture, we will develop convergence guarantees for all of these algorithms, which clarify the impact of:
  - Bias-correction
  - The step-size and gradient-noise
  - Network connectivity
  - Heterogeneity of local objectives

Conclusion

- Throughout this short-course, we developed a framework for developing and studying algorithms for decentralized multi-agent by using:
  - Stochastic gradient approximations
  - A decomposition of network evolution into centroid dynamics and the deviation from the centroid
- This general strategy for studying algorithms for multi-agent optimization and learning can be applied to many extensions. Here, we present three:
  - Multi-task and meta-learning
  - Compressed learning
  - Private learning over networks

Multi-Task Learning

Up to this point, in the context of distributed learning algorithms, we have exclusively focused on consensus optimization problems of the form:

$$w^o \triangleq \arg\min_{w\in\mathbb{R}^M} \sum_{k=1}^{K} J_k(w) \tag{1}$$

Since the resulting model w^o is unaffected by normalization, it can be viewed as the best average model for the aggregate objective $\frac{1}{K}\sum_{k=1}^{K} J_k(w)$. We can distinguish the consensus problem from non-cooperative problems:

$$w_k^o = \arg\min_{w_k\in\mathbb{R}^M} J_k(w_k) \Longleftrightarrow \mathcal{W}^o \triangleq \arg\min_{\mathcal{W}\in\mathbb{R}^{MK}} \sum_{k=1}^{K} J_k(w_k) \tag{2}$$

Since the local objectives are independent between agents, the locally optimal models w_k^o can be pursued in a non-cooperative manner by each individual agent using deterministic or stochastic gradient recursions.

Single-Task versus Multi-Task Learning

In a heterogeneous setting, we can distinguish two overarching network objectives:

- **Single-task learning:** Agents are interested in finding a common w^o:

$$w^o \triangleq \arg\min_{w \in \mathbb{R}^M} \sum_{k=1}^{K} J_k(w) \tag{3}$$

- **Multi-task learning:**[1] Agents are interested in finding a locally optimal model w_k^o, but would like to collaborate if it helps them in their own objective:

$$w_k^o = \arg\min_{w_k \in \mathbb{R}^M} J_k(w_k) \tag{4}$$

- Our algorithms and analysis thus far has focused on finding a common w^o.

[1] Multi-task learning is also known in some communities as "personalized learning".

Multi-task learning paradigms

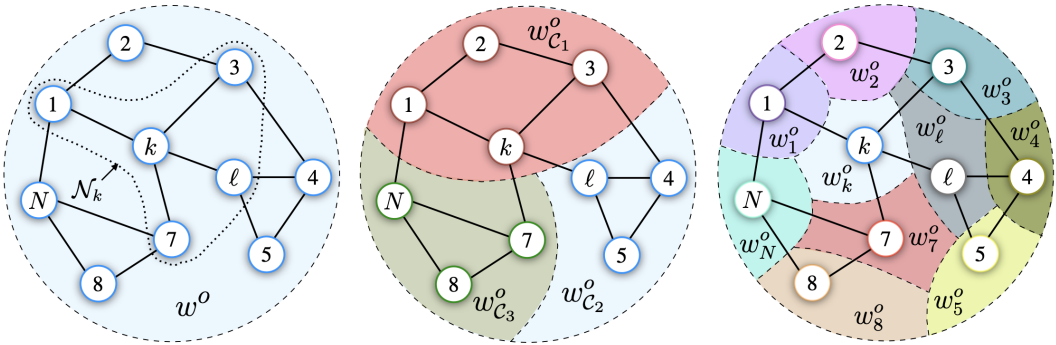


Figure: Examples of network learning paradigms. (Left) Single-task network. (Middle) Clustered multitask network. (Right) Multitask network.

Parametric multi-task learning

Parametric multitask learning: Parametric approaches for multitask learning impose a prior on the relationship between objectives $J_k(w)$ or the optimal local models w_k^o. These priors are generally informed by domain knowledge, physics, or outside information, and drive the cooperation between agents. Parametric approaches for decentralized multitask learning will relax the aggregate optimization problem to:

$$\mathcal{W}_\eta^o = \arg \min_{\mathcal{W} \in \mathbb{R}^{MK}} \sum_{k=1}^{K} J_k(w_k) + \eta R(\mathcal{W}), \quad \text{s.t.} \;\; \mathcal{W} \in \Omega \tag{5}$$

Generic Structure of Parametric Multi-Task Learning Algorithms

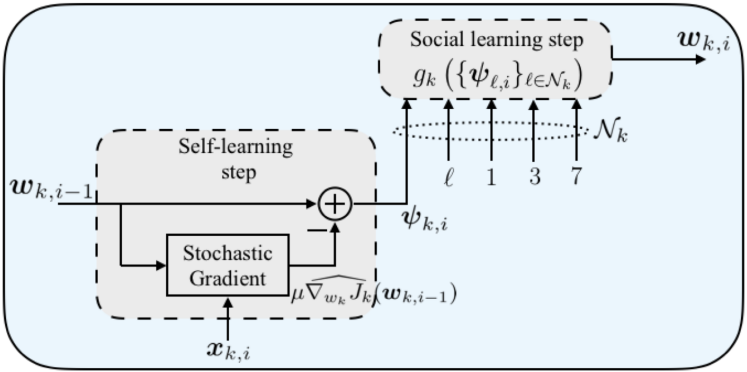


Figure: A generic structure for decentralized parametric mulit-task learning algorithms. It involves two steps, a self-learning step based on locally available data and a social learning step tuned to the underlying task-relatedness model.

Example: Multi-Task Learning with a Smoothness Prior

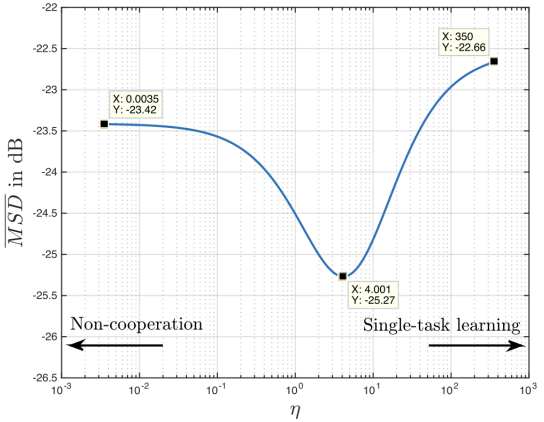
Recall the penalized approximation to the single-task problem, which we encountered in Lecture 4:

$$\arg\min_{\mathcal{W}} \sum_{k=1}^{K} J_k(w_k) + \frac{\eta}{2} \mathcal{W}^{\mathsf{T}} \mathcal{L} \mathcal{W} \tag{6}$$

In Lecture 4, where the objective was single-task optimization, we argued that this only results in a consensual solution when $\eta \to \infty$. Motivated by this observation, we set $\eta = \mu^{-1}$, which ensures that $\eta \to \infty$ for small step-sizes μ. If we don't want to enforce exact consensus, we can leave η as a hyperparameter.

- Algorithms for (6) are developed using the same techniques as we discussed in Lecture 4.

Performance of Multi-Task Learning with a Smoothness Prior



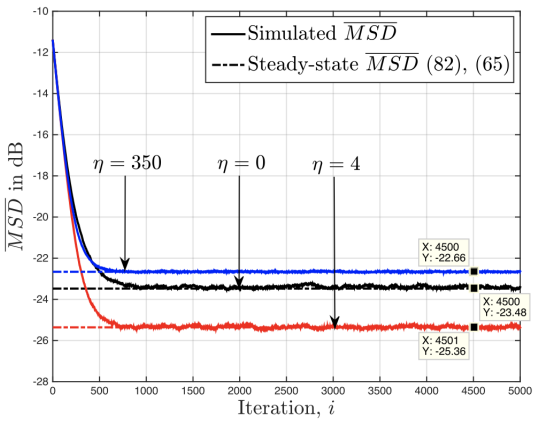


Figure: Performance of a diffusion-type multi-task learning algorithm for varying choices of η. Analytical expressions for this curve are available in [Nassif et al., 2020].

Application in Weather Prediction

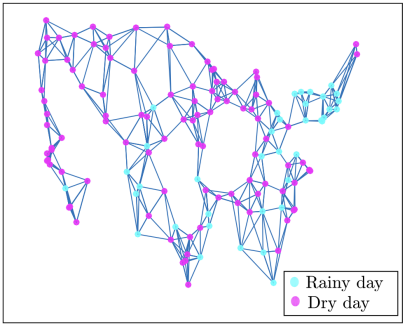
Suppose we would like to predict tomorrow's weather in different regions of the United States, based on a variety of different meteorological factors such as temperature and humidity. Whether it will rain on the following day can then be encoded in a binary class variables $\gamma_k = \pm 1$, and the available measurements in a feature vector $\boldsymbol{h}_k$. We may then formulate a rudimentary linear weather model by training a logistic regression classifier locally:

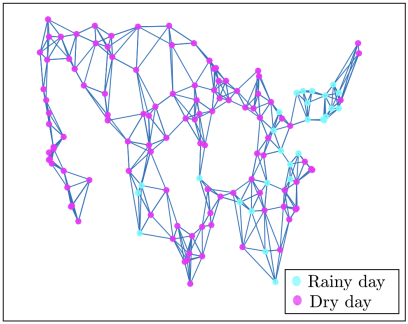
$$w_k^o \triangleq \arg\min_{w_k} \mathbb{E} \ln\left(1 + e^{-\gamma_k \boldsymbol{h}_k^{\mathsf{T}} w_k}\right) + \frac{\rho}{2}\|w_k\|^2 \tag{7}$$

Since weather patterns are likely to be similar in nearby geographical regions, we may also envision a distributed learning schemes, where we encourage smoothenss in the local weather models w_k^o, resulting in:

$$\mathcal{W}_\eta^o \triangleq \arg\min_{\mathcal{W}} \sum_{k=1}^{K} \left(\mathbb{E} \ln\left(1 + e^{-\gamma_k \boldsymbol{h}_k^{\mathsf{T}} w_k}\right) + \frac{\rho}{2}\|w_k\|^2\right) + \frac{\eta}{2}\,\mathcal{W}^{\mathsf{T}}\,\mathcal{L}\,\mathcal{W} \tag{8}$$

Application in Weather Prediction





	$\eta = 0$	$\eta = 10$	$\eta = 45$	$\eta = 100$	$\eta = 1000$	$\eta = \mu^{-1}$
prediction error	0.309	0.232	**0.225**	0.226	0.228	0.232

Figure: Weather prediction using multitask learning with smoothness prior [Nassif et al., 2020]. (*Top left*) Actual occurrence of rain. (*Top right*) Predicted occurrence of rain. (*Bottom*) Prediction accuracy as a function of the regularization parameter η of (6).

Subspace Constrained Multi-Task Learning

An alternative model-based setting may be one where tasks are not necessarily smooth over the graph, but instead linearly related, i.e., $\mathcal{W} \in \mathrm{Range}(\mathcal{U})$ for some $\mathcal{U}$.

$$\mathcal{W}^o = \arg\min_{\mathcal{W}} \sum_{k=1}^{K} J_k(w_k) \quad \text{s.t.} \quad \mathcal{W} \in \mathrm{Range}(\mathcal{U}), \tag{9}$$

where $\mathrm{Range}(\cdot)$ denotes the range space operator, and $\mathcal{U}$ is an $KM \times P$ full-column rank matrix with $P \ll KM$.

Note: We can actually recover the consensus problem (1) from (9) by setting $\mathcal{U} = \mathbb{1} \otimes I_M$. This is because for any x:

$$\mathcal{W} = (\mathbb{1} \otimes I_M)\, x \iff w_k = w_\ell \tag{10}$$

Example: Linearly-coupled optimization

Consider a setting where each agent k is estimating a subset of the global weight vector $w = [w^1, w^2, w^3]$, with potential overlap among agents.

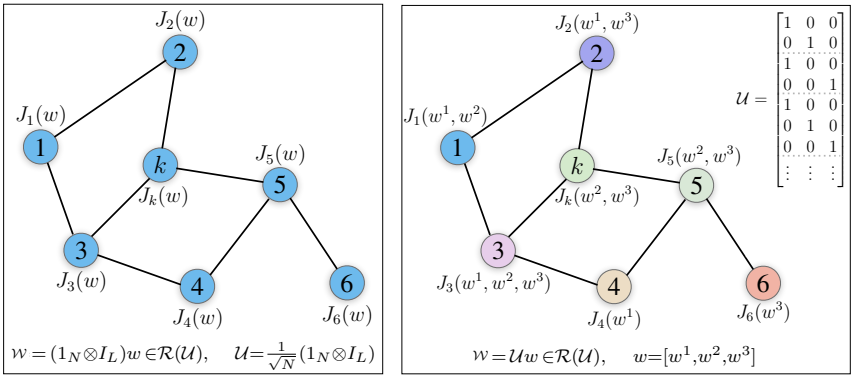
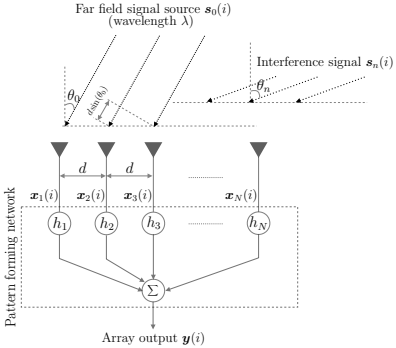


Figure: (*Left*) Consensus network and associated subspace matrix $\mathcal{U}$. (*Right*) Linearly coupled network and assocaited subspace matrix $\mathcal{U}$.

Application: Decentralized Beamforming



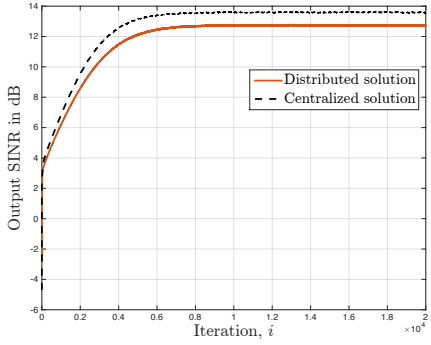


Figure: *(Left)* Uniform linear array of K antennas. *(Right)* Comparison of output SINR [Nassif et al., 2020].

Non-Parametric Multi-Task Learning via Meta-Learning

Instead of directly modeling the relationship between tasks w_k^o and w_ℓ^o, in model-agnostic meta-learning one assumes that both one or several (stochastic) gradient step away from a common launch-model:

$$w_k^o \approx w^o - \mu \nabla_w Q(w^o; \boldsymbol{x}_k) \tag{11}$$

One then optimizes:

$$w^o \triangleq \arg\min_w \frac{1}{K} \sum_{k=1}^{K} \mathbb{E} Q(w - \mu \nabla_w Q(w; \boldsymbol{x}_k^1); \boldsymbol{x}_k^2) \tag{12}$$

to determine a common launch model w^o, which adapts quickly to other tasks w_k^o via one or several (stochastic) gradient steps.

Dif-MAML: Decentralized Meta-Learning

If we denote:

$$\overline{Q}(w;\boldsymbol{x}_k^1,\boldsymbol{x}_k^2) \triangleq Q(w-\mu\nabla_w Q(w;\boldsymbol{x}_k^1);\boldsymbol{x}_k^2) \tag{13}$$

then the optimization problem

$$w^o \triangleq \arg\min_w \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\overline{Q}(w;\boldsymbol{x}_k^1,\boldsymbol{x}_k^2) \tag{14}$$

is a *single-task* problem over the common launch model w, and can be pursued via any of the decentralized algorithms we have encountered so far. For example, using diffusion:

$$\begin{aligned}\boldsymbol{\phi}_{k,i} &= \boldsymbol{w}_{k,i-1} - \mu\nabla\overline{Q}(\boldsymbol{w}_{k,i-1};\boldsymbol{x}_{k,i}^1,\boldsymbol{x}_{k,i}^2) \\ &= \boldsymbol{w}_{k,i-1} - \mu\nabla Q(\boldsymbol{w}_{k,i-1}-\mu\nabla_w Q(\boldsymbol{w}_{k,i-1};\boldsymbol{x}_{k,i}^1);\boldsymbol{x}_{k,i}^2) \qquad (15)\\ \boldsymbol{w}_{k,i} &= \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \qquad (16)\end{aligned}$$

Application to ImageNet

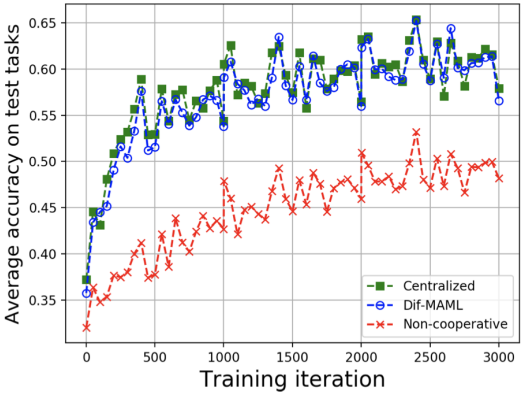


Figure: Performance of diffusion-based decentralized model-agnostic meta-learning on the ImageNet dataset [Kayaalp, Vlaski and Sayed, 2022].

Compressed Learning

Whenever messages are sent over a bandwidth-constrained communication channel, this will inevitably be associated with imperfections in the exchanged messages, which can be modelled as noise. We illustrate this on the diffusion algorithm again, though similar constructions apply to other decentralized algorithms we have encountered. Recall:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{17}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell,i} \tag{18}$$

Agents in this framework exchange intermediate estimates $\boldsymbol{\psi}_{k,i}$. In a digital communication setting, this is achieved by quantizing the vector $\boldsymbol{\psi}_{k,i}$ for a given bit-budget, and subsequently communicating the bit-representation of $\boldsymbol{\psi}_{k,i}$. We employ the notation $\boldsymbol{\mathcal{Q}}_k(\cdot)$ for the general quantization scheme employed by agent k, and can then write:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{19}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\mathcal{Q}}_\ell\left(\boldsymbol{\psi}_{\ell,i}\right) \tag{20}$$

Quantizer Conditions

Quantizer Conditions

The random quantization schemes $\boldsymbol{\mathcal{Q}}_k(\cdot)$ are unbiased, i.e.:

$$\mathbb{E}\left\{\boldsymbol{\mathcal{Q}}_k(\boldsymbol{\psi}_{k,i})|\boldsymbol{\psi}_{k,i}\right\} = \boldsymbol{\psi}_{k,i} \tag{21}$$

Furthermore, the variance of the quantization error satisfies the bound:

$$\mathbb{E}\left\{\left\|\boldsymbol{\psi}_{k,i} - \boldsymbol{\mathcal{Q}}_k(\boldsymbol{\psi}_{k,i})\right\|^2|\boldsymbol{\psi}_{k,i}\right\} \leq \beta_{k,q}^2\|\boldsymbol{\psi}_{k,i}\|^2 + \sigma_{q,k}^2 \tag{22}$$

Comparing the quantizer conditions (21)–(22) with our typical gradient noise conditions, we observe the same structure with a relative component proportional to the norm of the quantized quantity, and an absolute component.

Impact on Performance

We can interpret the quantization noise as a contributor to the stochastic gradient approximation:

$$\boldsymbol{s}_{k,i}^{\mathcal{Q}}(\boldsymbol{w}_{k,i-1}) = \boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i}) + \frac{1}{\mu}\left(\boldsymbol{Q}(\boldsymbol{\psi}_{k,i}) - \boldsymbol{\psi}_{k,i}\right) \tag{23}$$

It then follows from the results in Lecture 5 that:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \leq O\left(\frac{\mu \sum_{k=1}^K \sigma_k^2}{K^2\nu}\right) + O\left(\frac{\mu^{-1}\sum_{k=1}^K \overline{\sigma}_{q,k}^2}{K^2\nu}\right) + O(\mu^2) \tag{24}$$

In a homogenous setting we recover:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \leq O\left(\frac{\mu\sigma^2}{K\nu}\right) + O\left(\frac{\mu^{-1}\overline{\sigma}_q^2}{K\nu}\right) + O(\mu^2) \tag{25}$$

The results scales very poorly with the step-size μ.

Differential Quantization

For small step-sizes, however, models are updated slowly, and hence there is significant correlation between subsequent model estimates. This motivates the introduction of *differential quantization* schemes, which quantize the model updates instead of the models directly. As an example, we consider the algorithm from [Nassif et al, 2024]:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{26}$$

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \boldsymbol{\mathcal{Q}}_k(\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}) \tag{27}$$

$$\boldsymbol{w}_{k,i} = (1-\gamma)\boldsymbol{\phi}_{k,i} + \gamma \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \tag{28}$$

where

$$\boldsymbol{z}_{k,i} = (\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}) - \boldsymbol{C}_k(\boldsymbol{\psi}_{k,i} - \boldsymbol{\phi}_{k,i-1} + \boldsymbol{z}_{k,i-1}) \tag{29}$$

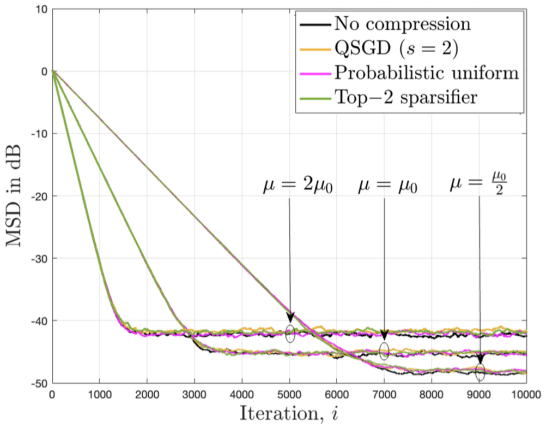
Performance with Differential Quantization and Error Feedback

Adopting similar analysis techniques to the ones we saw in lectures, we find:

$$\limsup_{i\to\infty} \mathbb{E}\| \boldsymbol{w}_{k,i} - w^o\|^2 = O\left(\frac{\mu\sigma^2}{\nu K}\right) + O(\sigma_q^2) \tag{30}$$

We can achieve $\sigma_q^2 = O(\mu^2)$ with $O(1)$-bits on average using a variable-rate quantizer (this requires proof [Nassif et al, 2022]).

Numerical Results



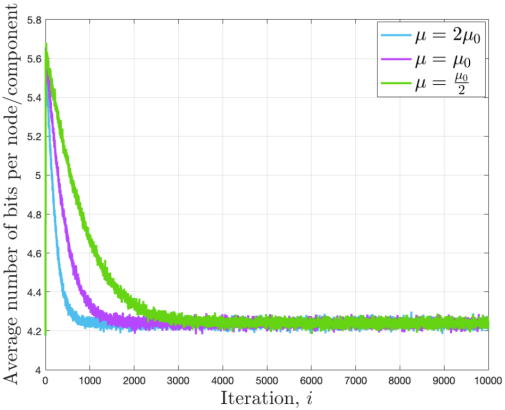


Figure: The proposed scheme matches that of an unquantized architecture despite the use of finite-bit quantization [Nassif et al, 2024]

Privacy

We begin with the observation that models and gradients can leak private information: For the mean squared error, for example, where:

$$J_k(w) = \mathbb{E}Q(w; \boldsymbol{x}_k) = \mathbb{E}\frac{1}{2}\left(\boldsymbol{\gamma}_k - \boldsymbol{h}_k^{\mathsf{T}} w\right)^2 \tag{31}$$

we have:

$$\widehat{\nabla J}_k^{\text{ele}}(w) = \nabla_w Q(w; \boldsymbol{x}_k) = -\boldsymbol{h}_k\left(\boldsymbol{\gamma}_k - \boldsymbol{h}_k^{\mathsf{T}} w\right) \tag{32}$$

For the logistic regression problem on the other hand, where:

$$J_k(w) = \mathbb{E}Q(w; \boldsymbol{x}_k) = \mathbb{E}\ln\left(1 + e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_k^{\mathsf{T}} w}\right) + \frac{\rho}{2}\|w\|^2 \tag{33}$$

we have:

$$\widehat{\nabla J}_k^{\text{ele}}(w) = \nabla_w Q(w; \boldsymbol{x}_k) = \rho w - \frac{\boldsymbol{h}_k}{1 + e^{\boldsymbol{\gamma}_k \boldsymbol{h}_k^{\mathsf{T}} w}} \tag{34}$$

Differential Privacy

Let us consider a collection of K agents indexed by k. Each agent has access to private data, which we model through the random variable $\boldsymbol{x}_k$. Any agent will have the option of participating in collaborative effort, which we describe generically as $\mathrm{Alg}\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\right)$. Consider also an alternative scenario, where an arbitrary agent, let us say without loss of generality, agent 1, refuses to participate in the learning protocol, and is replaced by some other agent $1'$ with different local data $\boldsymbol{x}_{1'} \in \mathcal{X}$ from a set of permissible local data sets $\mathcal{X}$.

ϵ-differential privacy

We say that an algorithm $\mathrm{Alg}\left(\cdot\right)$ is ϵ-differentially private, if it holds that:

$$\frac{f\left(\mathrm{Alg}\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\right)\right)}{f\left(\mathrm{Alg}\left(\boldsymbol{x}_{1'}, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\right)\right)} \leq e^{\epsilon} \tag{35}$$

for all $\boldsymbol{x}_{1'} \in \mathcal{X}$, where $f(\cdot)$ denotes the probability density function of the argument.

Example: Homogeneous Agents

Suppose the local data distributions are i.i.d, meaning $\boldsymbol{x}_1 \sim \boldsymbol{x}_2 \sim \ldots \sim \boldsymbol{x}_K \sim \boldsymbol{x}_{1'}$. It then follows that:

$$f\left(\mathrm{Alg}\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\right)\right) = f\left(\mathrm{Alg}\left(\boldsymbol{x}_{1'}, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\right)\right) \tag{36}$$

and hence the procedue is 0-differentially private. This essentially formalized the fact that if all information provided by agents is common knowledge, there is no privacy loss incurred by any given agent's partciipation.

Sensitivity of an Algorithm

ℓ_1-sensitivity

The sensitivity of $\mathrm{Alg}(\cdot)$ is defined as:

$$\Delta = \max_{\boldsymbol{x}_{1'} \in \mathcal{X}} \| \mathrm{Alg}\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\right) - \mathrm{Alg}\left(\boldsymbol{x}_{1'}, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\right) \|_1 \tag{37}$$

The Laplace Mechanism

Differential Privacy of the Laplace Mechanism

Suppose $\mathrm{Alg}(\cdot)$ has ℓ_1-sensitivity Δ, and define:

$$\mathrm{LAlg}(\cdot) = \mathrm{Alg}(\cdot) + \boldsymbol{v}_p \tag{38}$$

where $\boldsymbol{v}_p$ is a vector of suitable dimension M_v, where each entry follows the Laplace distribution:

$$f_{\boldsymbol{v}_p}(v) = \frac{1}{(2b_v)^{M_v}} e^{-\frac{\|v\|_1}{b_v}} \tag{39}$$

Then, $\mathrm{LAlg}(\cdot)$ is $\left(\frac{\Delta}{b_v}\right)$-differentially private.

Differential Privacy for Decentralized Learning

Motivated by this discussion, we can then consider the following variant of diffusion:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{40}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \boldsymbol{v}_{k,i} \tag{41}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell,i} \tag{42}$$

where $\boldsymbol{v}_{k,i}$ follows a Laplace distribution. Note again we can interpret this as augmenting the gradient noise to:

$$\boldsymbol{s}_{k,i}^{\text{priv}}(\boldsymbol{w}_{k,i-1}) = \boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) - \frac{1}{\mu}\boldsymbol{v}_{k,i} \tag{43}$$

Performance of the Naive Noise Addition

For small step-sizes this can result in serious amplification of the privacy noise component, yielding:

$$\sigma^2_{k,\text{priv}} = \sigma^2_k + \frac{\sigma^2_v}{\mu^2} \tag{44}$$

where σ^2_k denotes the absolute component of the gradient approximation $\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1})$ and σ^2_v denotes the variance of the Laplacian privacy noise $\boldsymbol{v}_{k,i}$. We can then conclude from results in Lecture 5, that the limiting performance of the privatized diffusion algorithm (for small step-sizes) will be given by:

$$\lim_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 = O(\mu\sigma^2) + O\left(\mu^{-1}\sigma^2_v\right) \tag{45}$$

Again we observe very poor scaling with the step-size.

Graph-Homomorphic Perturbations

The idea here will be to tune perturbations to the network topology in order to minimize their impact on learning performance while preserving privacy. The first step is to allow

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{46}$$

$$\boldsymbol{\psi}_{k\ell,i} = \boldsymbol{\phi}_{k,i-1} + \boldsymbol{v}_{k\ell,i} \tag{47}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell k,i} \tag{48}$$

Graph-Homomorphic Perturbations

For the network centroid, we have:

$$
\begin{aligned}
\boldsymbol{w}_{c,i} &\triangleq \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{w}_{k,i} \\
&\overset{(48)}{=} \frac{1}{K}\sum_{k=1}^{K}\sum_{\ell=1}^{K}a_{\ell k}\boldsymbol{\phi}_{\ell,i} + \frac{1}{K}\sum_{k=1}^{K}\sum_{\ell=1}^{K}a_{\ell k}\boldsymbol{v}_{\ell k,i} \\
&= \frac{1}{K}\sum_{\ell=1}^{K}\left(\sum_{k=1}^{K}a_{\ell k}\right)\boldsymbol{\phi}_{\ell,i} + \frac{1}{K}\sum_{\ell=1}^{K}\sum_{k=1}^{K}a_{\ell k}\boldsymbol{v}_{\ell k,i} \\
&\overset{(a)}{=} \frac{1}{K}\sum_{\ell=1}^{K}\boldsymbol{\phi}_{\ell,i} + \frac{1}{K}\sum_{\ell=1}^{K}\sum_{k=1}^{K}a_{\ell k}\boldsymbol{v}_{\ell k,i} \\
&\overset{(46)}{=} \boldsymbol{w}_{c,i-1} - \frac{\mu}{K}\sum_{\ell=1}^{K}\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) + \frac{1}{K}\sum_{\ell=1}^{K}\sum_{k=1}^{K}a_{\ell k}\boldsymbol{v}_{\ell k,i}
\end{aligned}
\tag{49}
$$

Graph-Homomorphic Perturbations

Graph-Homomorphic Perturbations

A set of perturbations $\boldsymbol{v}_{\ell k,i}$ is homomorphic for the the graph defined by the adjacency matrix $A \triangleq [a_{\ell k}]$ if it holds with probability one that:

$$\frac{1}{K}\sum_{\ell=1}^{K}\sum_{k=1}^{K} a_{\ell k}\boldsymbol{v}_{\ell k,i} = 0 \tag{50}$$

Let each agent ℓ sample independently from the Laplace distribution $\boldsymbol{v}'_{\ell,i} \sim \mathrm{Lap}\,(0, b_v)$ with variance $\sigma_v^2 = 2b_v^2$. Then, the construction:

$$\boldsymbol{v}_{\ell k,i} = \begin{cases} \boldsymbol{v}'_{\ell,i}, & \text{if } k \in \mathcal{N}_\ell \text{ and } k \neq \ell, \\ -\frac{1-a_{\ell\ell}}{a_{\ell\ell}}\boldsymbol{v}'_{\ell,i}, & \text{if } k = \ell. \end{cases} \tag{51}$$

is homomorphic for the graph described by the symmetric adjacency matrix $A = A^{\mathsf{T}}$.

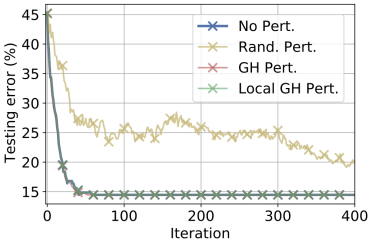
Perturbationce of Diffusion Algorithm with Graph-Homomorphic Perturbations

The fact that the privacy noise is cancelled in the centroid subspace allows us to obtain improved performance:

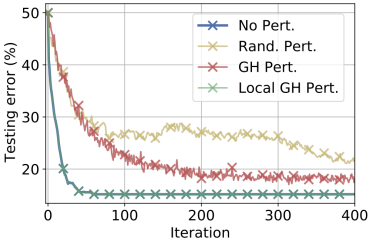
$$\lim_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 = O(\mu\sigma^2) + O\left(\sigma_v^2\right) \tag{52}$$

Complete cancellation does not occur, because gradients are still evaluated at individual iterates. But the improvement is compared to $O\left(\mu^{-1}\sigma_v^2\right)$ is still substantial for small step-sizes. Further improvement is possible with secure aggregation techniques.

Performance



(a) Centroid testing error



(b) Average individual testing error

Figure: Performance of privatized variants of the diffusion algorithm. Taken from [Rizk, Vlaski, Sayed 2023].

References and Further Reading

- General references and surveys:
  - A. H. Sayed, *Inference and Learning from Data*, Cambridge University Press, 2022.
  - A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, July 2014.
  - A. H. Sayed, "Adaptive Networks," in *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460-497, 2014.
  - S. Vlaski, S. Kar, A. H. Sayed and J. M. F. Moura, "Networked Signal and Information Processing: Learning by multiagent systems," in *IEEE Signal Processing Magazine*, vol. 40, no. 5, pp. 92-105, July 2023,
- Multi-task and meta-learning learning over networks:
  - R. Nassif, S. Vlaski, C. Richard, J. Chen and A. H. Sayed, "Multitask Learning Over Graphs: An Approach for Distributed, Streaming Machine Learning," in *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14-25, May 2020.
  - M. Kayaalp, S. Vlaski and A. H. Sayed, "Dif-MAML: Decentralized Multi-Agent Meta-Learning," in *IEEE Open Journal of Signal Processing*, vol. 3, pp. 71-93, 2022.

References and Further Reading

- Compressed learning:
  - R. Nassif, S. Vlaski, M. Carpentiero, V. Matta, M. Antonini and A. H. Sayed, "Quantization for Decentralized Learning Under Subspace Constraints," in *IEEE Transactions on Signal Processing*, vol. 71.
  - R. Nassif, M. Carpentiero, S. Vlaski, V. Matta, and A. H. Sayed, "Differential Error Feedback for Communication-Efficient Decentralized Optimization", to appear in *Proc. of IEEE SAM*, Corvallis, US, 2024.
- Private learning:
  - C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Found. Trends Theor. Comput. Sci., vol. 9, no. 3–4, pp. 211–407, Aug. 2014.
  - S. Vlaski and A. H. Sayed, "Graph-Homomorphic Perturbations for Private Decentralized Learning," *Proc. of IEEE ICASSP*, Toronto, ON, Canada, 2021.