In distributed optimization the objective function is given by,
\[
\min_{x \in \mathcal{X}} \sum_{i=1}^{N} f_i(x),
\]
with \( \mathcal{X} \subset \mathbb{R}^d \) a closed convex constraint set and \( f_i \) a convex function and \( N \) is the total number of nodes in the system. Each node \( i \) has access to its local objective function \( f_i \). The communication structure is defined through the underlying communication graph \( G := (V,E) \), where \( V \) and \( E \) are the vertices and edges, resp. The matrix \( A \) represents the communication graph and is assumed to be doubly stochastic.

Optimizing the objective amounts to finding a solution such that:
1. Consensus holds: \( \bar{x} = \bar{\bar{x}} \)
2. Optimality holds: \( \sum_{i=1}^{N} \nabla f_i(\bar{x}) = 0 \).

Can we find an algorithm such that both of these objectives are achieved?

The setting

A standard interacting stochastic mirror descent (ISMD) algorithm for estimating the minimizer is,
\[
\begin{align*}
\dot{x}_t &= -\eta \nabla \Phi^*(x_t) - \sigma dB_t, \\
\dot{v}_t &= \nabla \Phi^*(x_t)
\end{align*}
\]
where
\[
\nabla \Phi_i(x_t) := \nabla f_i \circ \nabla \Phi^*(x_t) \quad B_t := (B^1_t, \ldots, B^N_t)^T \quad \nabla \Phi(x_t) = (\nabla \Phi_1(x_t)^T, \ldots, \nabla \Phi_N(x_t)^T)^T
\]
Will this algorithm converge to consensus and optimality?

Algorithm 1.

Under the assumptions of smoothness and convexity of \( f_i \) it holds,
\[
\frac{1}{T} \int_0^T \mathbb{E} \left[ (f(x_t) - f(x^*)) \right] dt \leq C_1 + C_2 \sigma^2 + C_3 \frac{\sigma^2}{\eta} + C_4 \frac{\sigma^2}{\eta N} + C_5 \frac{\sigma}{\sqrt{T}},
\]
so that:
1. Imposing a small learning rate slows down convergence but allows to converge closer to the optimum if the noise is small or number of particles is big.
2. Imposing a high interaction strength allows to converge closer to the optimum.

Exact convergence is this not achieved due to:
• An additional term arising from the noise,
• An additional term arising from the gradients. This term can only be mitigated by imposing a small learning rate, but this slows down convergence!

How can we mitigate this?

The exact algorithm converges a lot closer and faster to the optimum. Using a small learning rate or high interaction can help converge closer too.

Algorithm 2.

We propose an exact algorithm:
\[
\begin{align*}
\dot{x}_t &= -\eta \nabla \Phi^*(x_t) - \nabla f_i(x_t) dB_t + \sigma dB_t, \\
\dot{v}_t &= \nabla \Phi^*(x_t)
\end{align*}
\]
and note that \( \nabla^2 f_i(x_t) dx_t = d [\nabla f_i(x_t)], \) which when discretized yields the update \( \nabla f(x_{t+1}) = \nabla f(x_t). \)
So what is special here?
• This algorithm incorporates a form of history information.
• Before, the algorithm would be unstable if 1 and 2. was satisfied. Now it is stable.

Example 1. A linear system

The exact algorithm converges a lot closer and faster to the optimum. Using a small learning rate or high interaction can help converge closer too.

Example 1. A federated learning model.

Theoretically all should work in convex case. But what about the non-convex case where the model is a neural network? We see the exact algorithm performs good too.