Final project

Elisa Riccietti and Theo Mary

LIP-ENS Lyon

The context

Consider a large-scale data-fitting problem:

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} f_i(x) \tag{1}$$

for $f : \mathbb{R}^n \to \mathbb{R}$ and $n \ll m$.
In this context the limiting factor is thus the number of samples rather than the dimension of the variables space.

- Objective: Devise an extension of the multilevel schemes we have studied for the hierarchies of variables to this context and test it (use a hierarchy on the samples set).

Classical two-level algorithm: k-th iteration

Given x_k^h

1. Fine iteration: iteration on f^h.
   1.1 Compute a fine gradient step: $s_k^h = -\nabla f^h(x_k^h)$.
   1.2 Do a line search on f^h to choose α_k
   1.3 Set $x_{k+1}^h = x_k^h + \alpha_k s_k^h$.
2. Coarse iteration:
   2.1 Initialisation : $x_0^H = I_h^H x_k^h$
   2.2 Compute first order coherence $v^H = I_h^H \nabla f^h(x_k^h) - \nabla f^H(x_k^H)$
   2.3 Starting from x_0^H, apply p steps of a minimization algorithm with line-search on the problem

$$\min_{x^H} f^H(x^H) + (v^H)^T x^H$$

   getting x_p^H. Set $s_{p,k}^H = x_p^H - x_0^H$.
   2.4 Consider the step $I_H^h s_{p,k}^H$ and choose α_k via a line search on f^h
   2.5 Fine level update: $x_{k+1}^h = x_k^h + \alpha_k I_H^h s_{p,k}^H$

Classical multilevel algorithm with more levels

1. Ingredient: minimizer for a fixed level.
   Consider $OPT(x_0, f, v, p)$ a function that, starting from x_0, applies p iterations of a convergent algorithm to

   $$\min_x f(x) + v^T x$$

2. Given x_0^h, set $v^h = 0$. For $k = 0, \ldots,$ set

   $$x_{k+1}^h = ML(f^h, v^h, x_k^h)$$

   where $ML(f^h, v^h, x_k^h)$ is defined in the following slide.

Classical multilevel algorithm with more levels

1. If coarsest level $x_{k+1}^h = OPT(x_k^h, f^h, v^h, p_2)$
2. Fine iterations: $\bar{x}^h = OPT(x_k^h, f^h, v^h, p_1)$
3. Coarse iteration (recursion):
   3.1 Initialisation : $\bar{x}^H = I_h^H \bar{x}^h$
   3.2 Compute first order coherence
   $\bar{v}^H = I_h^H v^h + I_h^H \nabla f^h(x_k^h) - \nabla f^H(x_k^H)$
   3.3 $x_+^H = ML(f^H, \bar{v}^H, \bar{x}^H)$
   3.4 Compute the step $e^H = x_+^H - \bar{x}^H$ and $e^h = I_H^h e^H$
   3.5 Choose α_k via a line search such that $f^h(x_+^h) \leq f^h(\bar{x}^h)$

Test problems

- Choose your favorite dataset for binary classification with data (z_i, y_i) for $i = 1, \ldots, m$
- Consider a simple logistic model:

$$f_i = \frac{1}{N} \log(1 + e^{-y_i x^T z_i})$$

  with $(z_i, y_i) \in \mathbb{R}^n \times \{-1, 1\}$ and $x \in \mathbb{R}^n$.
- Consider a simple nonlinear least squares model with sigmoid loss:

$$f_i = \frac{1}{N} \left(y_i - \frac{1}{1 + e^{-y_i x^T z_i}} \right)$$

  with $(z_i, y_i) \in \mathbb{R}^n \times \{0, 1\}$ and $x \in \mathbb{R}^n$.

Task 1

- Implement a two-level algorithm.
- Define a suitable hierarchy (f^H?) for (**??**)
- Solve problem (**??**) with the test models and your two-level algorithm
- Compare it to a mini-batch version of SVRG (cf. course 3 and next slide).
- Be careful in your comparison! What is a fair measure for comparing the two algorithms?
- What is the difference between the two test problems?
- Optional: repeat with more than two-levels.

Minibatch SVRG

Requires: p frequency update parameter, α learning rate, initial guess $\tilde{x}_0$

1. Iterate for $s = 1, \dots$
   1.1 Set $\tilde{x} = \tilde{x}_{s-1}$
   1.2 Compute $\tilde{G} = \frac{1}{m} \sum_{i=1}^{m} \nabla f_i(\tilde{x})$
   1.3 $x_0 = \tilde{x}$
   1.4 Iterate for $k = 1, \dots, p$
      1.4.1 Choose randomly $S_k \subset \{1, \dots, m\}$
      1.4.2 $x_k = x_{k-1} - \alpha(\frac{1}{|S_k|} \sum_{i \in S_k} \nabla f_i(x_{k-1}) - \frac{1}{|S_k|} \sum_{i \in S_k} \nabla f_i(\tilde{x}) + \tilde{G})$
   1.5 Set $\tilde{x}_s = x_p$

Task 2: Analyse a step of mini-batch SVRG and a step of the two-level strategy (when the coarse model is employed). Derive a link between the two of them.