Introduction

We study the problem of minimizing the average of a large number of $1/\gamma$-smooth convex functions penalized with a 1-strongly convex regularizer.

$$\min_{w \in \mathbb{R}^d} P(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(w_i) + \lambda g(w). \tag{P}$$

Each $a_i \in \mathbb{R}^d$ and we write $A = [a_1, \ldots, a_n] \in \mathbb{R}^{d \times n}$. Let $g^*$ and $(\phi^*_i)$ be the Fenchel conjugate functions of $g$ and $(\phi_i)$, respectively. In the case of $g$, for instance, we have $g^*(z) = \sup_{w \in \mathbb{R}^d} \langle w, z \rangle - g(w)$.

The (Fenchel) dual problem of (P) can be written as:

$$\max_{\alpha \in \mathbb{R}^n} D(\alpha) := \frac{1}{n} \sum_{i=1}^{n} -\phi_i^*(-a_i) - \lambda g^*\left(\frac{1}{n} A \alpha\right). \tag{D}$$

The Algorithm

**Sampling $\hat{S}$:** A random subset of $\{1, 2, \ldots, n\}$ such that $\forall i: \text{Prob}(i \in \hat{S}) > 0$ and $\text{Prob}(\hat{S} = \emptyset) = 0$.

**Algorithm 1:** SDNA Algorithm

1. **Initialization:** $\alpha^0 \in \mathbb{R}^n$; $\hat{\alpha}^0 = \frac{1}{\lambda \sigma} A \alpha^0$
2. for $k = 0, 1, 2, \ldots$ do
3. Primal update: $w^k = \nabla g^*(\hat{\alpha}^k)$
4. Generate a random set of blocks $S_k \sim \hat{S}$
5. Compute:
   $$\Delta \alpha_k = \arg \min_{h : h \in \mathbb{R}^n} \langle A^T w^k, I_{S_k} h \rangle + \frac{\lambda}{2} h^\top X_{S_k} h + \sum_{i \in S_k} \phi_i^*(-a_i^k - h_i)$$
6. Dual update: $\alpha^{k+1} := \alpha^k + (\Delta \alpha_k)_{S_k}
7. $\alpha^{k+1} := \alpha^k + \frac{1}{\lambda \sigma} \sum_{i \in S_k} \Delta \alpha_i a_i$
8. end for

Where $X = A^T A$ and $X_{S_k}$ is the matrix obtained from $X$ retaining elements $X_{ij}$ for which both $i, j \in S_k$ and zeroing out all other elements.

Iteration Complexity of SDNA

**Theorem:** Let $\hat{S}$ be a uniform sampling and let $\tau := \frac{\theta(\hat{S})}{\theta(S)}$. The output sequence $\{w^k, \alpha^k\}_{k \geq 0}$ of Algorithm 1 satisfies:

$$\text{E}[P(w^k) - D(\alpha_k)] \leq \left(1 - \sigma \frac{k}{\theta(S)} \right) \left( D(\alpha^*) - D(\alpha^0) \right),$$

where $\sigma := \frac{\min(1, 1 + \frac{1}{n})}{\min(1, 1 + 1/n)}$, $s_1 = \lambda \min\left( \left( \frac{1}{\tau \lambda} \text{E}[((A^T A)_{S_k} + I)^{-1}] \right)^{-1} \right)$ and $v \in \mathbb{R}^n_{++}$ is a vector satisfying:

$$\text{E}[(A^T A)_{S_k}] \preceq \text{diag}(p) \cdot \text{diag}(v). \tag{1}$$

Comparison with Mini-Batch SDCA

**Algorithm 2:** Minibatch SDCA

1. **Parameters:** uniform sampling $\hat{S}$, vector $v \in \mathbb{R}^n_{++}$
2. **Initialization:** $\alpha^0 \in \mathbb{R}^n$; set $\hat{\alpha}^0 = \frac{1}{\lambda \sigma} A \alpha^0$
3. for $k = 0, 1, 2, \ldots$ do
4. Primal update: $w^k = \nabla g^*(\hat{\alpha}^k)$
5. Generate a random set of blocks $S_k \sim \hat{S}$
6. Compute for each $i \in S_k$
   $$h_i^k = \arg \min_{h_i} (\alpha_i^k w_i^k) + \frac{\lambda}{2} h_i^2 + \phi_i^*(-\alpha_i^k - h_i)$$
7. Dual update: $\alpha^{k+1} := \alpha^k + \sum_{i \in S_k} h_i^k e_i$
8. Average update: $\hat{\alpha}^{k+1} := \hat{\alpha}^k + \frac{1}{n} \sum_{i \in S_k} h_i^k a_i$
9. end for

**Theorem:** If (1) holds, then the output sequence $\{w^*, \alpha^*\}_{k \geq 0}$ of Algorithm 2 satisfies:

$$\text{E}[P(w^k) - D(\alpha^k)] \leq \left(1 - \theta(\hat{S}) \right) \left( D(\alpha^*) - D(\alpha^0) \right).$$

Moreover, $\theta(\hat{S}) \leq \sigma$.

Numerical Experiments

Comparison of SDNA and SDCA for minibatch sizes $\tau = 1, 32, 256$ on a real (left) and synthetic (right) dataset. The methods coincide for $\tau = 1$.

Time it takes for SDNA and SDCA to process a single epoch as a function of the minibatch size $\tau$.

References