Nonconvex Variance Reduced Optimization with Arbitrary Sampling

Samuel Horváth1 Peter Richtárik1,2,3

1 KAUST 2 University of Edinburgh 3 Moscow Institute of Physics and Technology

The Problem

\[ \min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \]  

\[ \bullet f_i \text{ is } L_i \text{-smooth but non-convex} \]
\[ \bullet n \text{ is big} \]

 Arbitrary Sampling

\[ \text{Sampling: a random set-valued mapping } S \text{ with values being subsets of } [n] := \{1, 2, \ldots, n\}. \]
\[ \text{A sampling is used to generate minibatches in each iteration.} \]
\[ \text{Probability matrix associated with sampling } S : \]
\[ P_{ij} = \text{Prob}(i,j) \subseteq S \]
\[ \text{Probability vector associated with sampling } S : \]
\[ p = (p_1, \ldots, p_n), \quad p_i = \text{Prob}(i \in S) \]
\[ \text{Minibatch size: } b = E[|S|] \text{ (expected size of } S) \]
\[ \text{Proper sampling: Sampling for which } p_i > 0 \text{ for all } i \in [n] \]
\[ \text{“Arbitrary sampling” = any proper sampling} \]

Main Contributions

\[ \text{We develop arbitrary sampling variants of 3} \]
\[ \text{popular variance-reduced methods for solving the non-convex problem (1): SVRG [1], SAGA [2], SARAH [3].} \]
\[ \text{We are able to compute the optimal sampling out} \]
\[ \text{of all samplings of a given minibatch size. This is the first time an optimal minibatch sampling} \]
\[ \text{was computed (from the class of all samplings).} \]
\[ \text{We design importance sampling & approximate importance sampling for minibatches, which} \]
\[ \text{vastly outperform standard uniform minibatch strategies in practice.} \]

Key Lemma

Let \( \zeta_1, \zeta_2, \ldots, \zeta_n \) be vectors in \( \mathbb{R}^d \) and let \( \bar{\zeta} := \frac{1}{n} \sum_{i=1}^{n} \zeta_i \) be their average. Let \( S \) be a proper sampling. Let \( v = (v_1, \ldots, v_n) > 0 \) be such that
\[ P = pp^T \leq \text{Diag}(p_1, p_2, \ldots, p_n) \]  

Then
\[ E\left[\sum_{i \in S} \frac{\zeta_i - \bar{\zeta}}{p_i}\right]^2 \leq \frac{1}{n^2} \sum_{i=1}^{n} v_i \|\zeta_i\|^2. \]

Whenever (2) holds, it must be the case that
\[ v_i \geq 1 - p_i. \]

Optimal Sampling & Superlinear Speedup

\[ \bullet \text{Under our analysis, the independent sampling } S^o \text{ defined by} \]
\[ p_i = \begin{cases} \frac{b + k - n}{\sum_{j=1}^{k} L_j}, & \text{if } i \leq k \in \mathbb{N} \\ 1, & \text{if } i > k \end{cases} \]

is optimal, where \( k \) is the largest integer satisfying \( 0 < b + k - n \leq \sum_{j=1}^{k} L_j \).
\[ \bullet \text{All 3 methods enjoy superlinear speed in } b \text{ up to the minibatch size} \]
\[ b_{\max} := \max\{b \mid b L_i \leq \sum_{j=1}^{k} L_j\}. \]

Numerical Results

<table>
<thead>
<tr>
<th>Alg</th>
<th>Uniform sampling</th>
<th>Arbitrary sampling [NEW]</th>
<th>S^o (Best Sampling) [NEW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVRG</td>
<td>[ n + \frac{2\max_{i \in [n]}(1 + \log(\frac{\alpha}{\epsilon}))}{\epsilon} ]</td>
<td>[ n + \frac{(1 + \log^2(\frac{\epsilon}{\alpha}))}{\epsilon} ]</td>
<td>[ n + \frac{(1 + \log^2(\frac{\epsilon}{\alpha}))}{\epsilon} ]</td>
</tr>
<tr>
<td>SAGA</td>
<td>[ n + \frac{2\max_{i \in [n]}(1 + \log(\frac{\alpha}{\epsilon}))}{\epsilon} ]</td>
<td>[ n + \frac{(1 + \log^2(\frac{\epsilon}{\alpha}))}{\epsilon} ]</td>
<td>[ n + \frac{(1 + \log^2(\frac{\epsilon}{\alpha}))}{\epsilon} ]</td>
</tr>
<tr>
<td>SARAH</td>
<td>[ n + \frac{\log^2(\frac{\alpha}{\epsilon})}{\epsilon} ]</td>
<td>[ n + \frac{(1 + \log^2(\frac{\epsilon}{\alpha}))}{\epsilon} ]</td>
<td>[ n + \frac{(1 + \log^2(\frac{\epsilon}{\alpha}))}{\epsilon} ]</td>
</tr>
</tbody>
</table>

Constants: \( L_{\max} = \max\{L_i\}, \quad L = \frac{1}{n} \sum_{i=1}^{n} L_i \quad c_1, c_2, c_3 = \text{universal constants} \quad \alpha := \frac{\delta}{L_0} \sum_{i=1}^{k} L_i \epsilon} \]

SVRG with Arbitrary Sampling

Algorithm 1: SVRG
\[ x^0 = x_0, \quad g^0 = \nabla f(x^0), \quad M = [T/m]; \]
for \( s = 0 \) to \( M - 1 \) do
\[ x_{s+1} = x_s - \eta_{s} g_{s+1} = \frac{1}{\eta_{s}} \nabla f(x_{s+1}) \]
for \( t = 0 \) to \( m - 1 \) do
\[ \text{Draw a random subset (minibatch) } S_t \sim S \]
\[ \zeta_{s+1} = \sum_{i \in S_t} \frac{1}{p_i} (\nabla f_i(x_{s+1}) - \nabla f_i(x_{s+1}^0)) + g_{s+1} \]
\[ x_{s+1} = x_{s+1} - \eta_{s+1} \zeta_{s+1} \]
end
\[ \bar{x}_{s+1} = \frac{1}{m} \sum_{s=0}^{m-1} x_{s+1} \]
end

Output: Iterate \( x_{s+1} \) chosen uniformly random from \( \{x_{s+1}\}_{s=0}^{M-1} \)

References