Safe Adaptive Importance Sampling

sebastian.stich@epfl.ch, anant.raj@tuebingen.mpg.de, martin.jaggi@epfl.ch

**Summary**

**Optimal adaptive sampling (full information)**
The performance of stochastic optimization algorithms like Stochastic Gradient Descent (SGD) and Coordinate Descent (CD) crucially depends on the sampling distribution. The progress is maximized for full information optimal adaptive sampling $p^{opt}$, but this distribution requires knowledge of full gradient information and is therefore unamenable in practice.

**Safe bounds: a relaxation**

CD setting:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \Rightarrow \quad \|x_i - \nabla f_i(x_i)\| \leq \|u_i\|$$

SGD setting:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x_i) \quad \Rightarrow \quad \|x_i - \nabla f_i(x_i)\| \leq \|u_i\|$$

**Optimal adaptive sampling (limited information)**
The optimal adaptive sampling with respect to the bounds $\ell \leq u$ is the solution of an optimization problem and can efficiently be computed. We propose to use limited information optimal adaptive sampling $p^{opt}$.

For any bounds $\ell \leq u$, the proposed sampling $p^{opt}$ is provably better than importance sampling.

**Example: Coordinate Descent**

(also applicable to SGD)

**Alg. 1: Optimal sampling (too expensive)**
- Compute: $\nabla f(x_i)$
- $\ell_i = u_i = \|\nabla f(x_i)\|$
- Iteration complexity: $T(\nabla f(x_i))$

**Alg. 2: Proposed sampling**
- Compute: $\nabla f(x_i)$
- Update $\ell_i \leq \|\nabla f(x_i)\| \leq u_i$
- Iteration complexity: $T(\nabla f(x_i)) + O(n \log n)$

**Alg. 3: Importance sampling (slower convergence)**
- Compute: $\nabla f(x_i)$
- $\ell_i = u_i = \infty$
- Iteration complexity: $T(\nabla f(x_i))$

The stepsize $\frac{1}{n}$ and the sampling $p_i$ maximize the expected one step progress (in the worst case):

$$f(x_i) - f(x_{i+1}) \geq \frac{\|\nabla f_i(x_i)\|^2}{\ell_i + \|\nabla f_i(x_i)\|^2}$$

$$(y_i, p_i) := \arg \max_{y \in \mathbb{R}^d, \, p \in \mathbb{P}} \min_{1 \leq i \leq n} \mathbb{E}_{x_i \sim p}[f(x_i; y_i, p)]$$

**Assumptions:** $f: \mathbb{R}^n \rightarrow \mathbb{R}$ convex; coordinate-wise Lipschitz: $\|\nabla f(x + \gamma a_i) - \nabla f(x)\| \leq L_i |\gamma|$, $\forall x \in \mathbb{R}^n, \gamma \in \mathbb{R}, i = 1, \ldots, n$. Define: $L = \text{diag}(L_1, \ldots, L_n)$.

**Details**

- Trivial values $\ell_i = 0$ and $u_i = \infty$ are admissible, but more accurate bounds give better speed-up.
- Updating the bounds can be delegated to a dedicated worker in a distributed setting.

**Special cases:**
- $\ell = u$: optimal sampling (full information)
- $\ell = 0$, $u = \infty$: uniform sampling (no information)

**Principal example:** $f(x) = \frac{1}{2} x^T A x - b^T x$.

**CD setting:** ($A \in \mathbb{R}^{d \times n}$)

$$\nabla f(x + \gamma a_i) - \nabla f(x) = (a_i, a_i) \gamma_{x_i} \quad \forall i \neq k$$

**SGD setting:** ($A \in \mathbb{R}^{n \times d}$, $f(x) = \sum_{i=1}^d f_i(a_i^T x)$)

$$\nabla f_i(x + \gamma a_i) - \nabla f_i(x) = (a_i, a_i) \gamma_{x_i} \quad \forall i \neq k$$

**Practical example: $\ell = \frac{1}{2}$, $u = \frac{3}{2}$, $\nabla f(x_i) = \frac{2}{3}$ and $L_1 = L_2 = 2$. Then $\mathbb{E}_{x_i \sim \text{uniform}}[f(x_{i+1})] < \max_{p \in \mathcal{P}} \mathbb{E}_{x_i \sim p}[f(x_{i+1})]$.**

**Experiments**

**Iterations:** Coordinate Descent on rvcl

<table>
<thead>
<tr>
<th>$f(x)$</th>
<th>Uniform</th>
<th>Proposed</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x)$</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Epochs</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x)$</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**Clock time:** Coordinate Descent on real-sim

<table>
<thead>
<tr>
<th>$f(x)$</th>
<th>Uniform</th>
<th>Proposed</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x)$</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x)$</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
</tr>
</tbody>
</table>

**Open Problems and Future Work**

- The approach works very well for the CD setting, for SGD the advantage was less pronounced in the experiments, can this be explained by theory?
- Relax the strict conditions on $\ell$ and $u$
- Reduce complexity to $O(\min\{d, n\})$ (for both: updating the bounds $\ell$, $u$, and computing $p^{opt}$).
- The approach might be transferable to other domains, for instance active learning.