Large-Scale Optimization

L6. STOCHASTIC GRADIENT DESCENT - 2
Today:

Classical SA/SGD: strongly convex
Robust SA/SGD: (weakly) convex
Stochastic Approximation

\[ \min_{x \in X \subset \mathbb{R}^n} f(x) = \mathbb{E}[F(x, \xi)] \]

Using noisy information about the function

Initial ideas:

- Robbins & Monro, 1951
- Kiefer & Wolfowitz, 1952
Vladimir Vapnik

Father of “statistical learning theory”
  • Vapnik-Chervonenkis (VC) dimension
  • Support vector machines

In his book “The Nature of Statistical Learning Theory” (1999),
  • “The stochastic approximation principle is, however, too wasteful: It uses one element of the training data per step…”
Robust Stochastic Approximation Approach to Stochastic Programming, SIOPT, 2009

Arkadi Nemirovskii, Anatoli Juditsky, Guanghui Lan, Alexander Shapiro
Classical SA/SGD

For $j=1,2,...$

- Choose a random vector $\xi_j$
- Choose a stochastic subgradient $G(x_j, \xi_j) \in \partial F(x_j, \xi_j)$
- Take a step $x_{j+1} = \prod_x (x_j - \gamma_j G(x_j, \xi_j))$

**Dependencies:**

$x_j$ depends on $\xi_{[j-1]} = \{\xi_1, \xi_2, \ldots, \xi_{j-1}\}$
Classical SA/SGD

\[
\min_{x \in \mathbb{R}^n} f(x) = \mathbb{E}[F(x, \xi)]
\]

\[
x_{j+1} = \prod_{x} (x_j - \gamma_j G(x_j, \xi_j))
\]

Analysis in Nemirovski et al., 2009, pp1577-1578

Assumptions:

\[
\mathbb{E}[\|G(x, \xi)\|_2^2] \leq M^2, \ \forall x \in X
\]
Strong Convexity with Subgradients

\[ f(y) \geq f(x) + g(x)^T(y - x) + \frac{c}{2} \|y - x\|^2_2, \quad \forall x, y \in X \]

\[ f(x) \geq f(y) + g(y)^T(x - y) + \frac{c}{2} \|x - y\|^2_2 \]

\[ (g(y) - g(x))^T(y - x) \geq c \|y - x\|^2_2 \]

\[ (g(x_j) - g(x^*))^T(x_j - x^*) \geq c \|x_j - x^*\|^2_2 \]
(Fundamental) Optimality Condition

$$(x - x^*)^T g(x^*) \geq 0, \ \forall x \in X, \exists g(x^*) \in \partial f(x^*)$$
From two results:

- Strong convexity

\[(g(x_j) - g(x^*))^T (x_j - x^*) \geq c\|x_j - x^*\|^2\]

- Optimality condition

\[(x - x^*)^T g(x^*) \geq 0, \quad \forall x \in X, \exists g(x^*) \in \partial f(x^*)\]

\[g(x_j)^T (x_j - x^*) \geq c\|x_j - x^*\|^2\]
Classical SGD: Convergence

Stepsize: \( \gamma_j = \frac{\theta}{j}, \quad \theta > \frac{1}{2c} \) \hspace{1cm} \text{Diminishing, square summable}

Convergence in iterates: \( \mathbb{E}[\|x_j - x^*\|^2] \leq Q(\theta) \frac{1}{j} \)

\[ Q(\theta) := \max\{\theta^2 M^2 (2c\theta - 1)^{-1}, \|x_1 - x^*\|^2\} \]

Convergence in obj. function values:

- When \( f \) is differentiable and has Lipschitz continuous gradients

\[ \mathbb{E}[f(x_j) - f(x^*)] \leq Q(\theta) \frac{L}{2j} \quad \to 0 \text{ as } j \to \infty \]
Check

How does the analysis change if:

• no projection onto X (unconstrained minimization)

• use full (non-stochastic) subgradient?
  – reverts to subgradient method, using 1/j stepsizes for strongly convex f (for which we had convergence but no rate)

• f is differentiable and use full gradient?
  – reverts to gradient descent, using 1/j stepsizes for strongly convex f (we didn’t use such stepsizes, and we had better choices leading to linear convergence)
Classical SGD: Weakness

With a wrong estimate of the strong convexity parameter $c$, the convergence could be very slow.

$\frac{1}{j}$ stepsizes do require strong convexity. If we apply $\frac{1}{j}$ stepsizes to weakly convex functions, convergence could be extremely slow.
Robust SGD

\[ x_{j+1} = \prod_x (x_j - \gamma_j G(x_j, \xi_j)) \]

\[ \mathbb{E}[\| G(x, \xi) \|^2_2] \leq M^2, \ \forall x \in X \]

\[ D = \max_{x \in X} \| x - x_1 \|_2 \]

Stepsizes

• Constant:
  \[ \gamma_t = \frac{D}{M \sqrt{N}} \]

• Decreasing:
  \[ \gamma_t = \frac{D}{M \sqrt{t}} \]
Robust SGD

First, what about $f(\text{best})$?

- Also in the classical SGD?
- This may not be useful since $f$ can be hard to compute due to expectation

Instead, we use an weighted average of iterates as a convergent object:

$$
\nu_t \geq 0, \sum_{t=i}^j \nu_t = 1
$$

$$
\tilde{x}_i^j = \sum_{t=i}^j \frac{\gamma_t}{\sum_{\tau=i}^j \gamma_\tau} x_t = \nu_j x_j + \frac{\sum_{\tau=i}^{j-1} \gamma_\tau}{\sum_{\tau=i}^j \gamma_\tau} \tilde{x}_{i}^{j-1}
$$
Robust SGD: Analysis

Nemirovski et al., pp1579-1580
Robust SGD: Convergence

Stepsize: \[ \gamma_t = \frac{\theta D}{M \sqrt{t}} \]

Convergence:
\[
\mathbb{E}[f(\tilde{x}_K^N) - f(x^*)] \leq \max\{\theta, \theta^{-1}\} \frac{DM}{\sqrt{N}} \left[ \frac{2N}{N - K + 1} + \frac{1}{2} \right]
\]

Constant (optimal)

\[
\mathbb{E}[f(\tilde{x}_K^N) - f(x^*)] \leq \max\{\theta, \theta^{-1}\} \frac{DM}{\sqrt{N}} \left[ \frac{2N}{N - K + 1} + \frac{1}{2} \right]
\]

Diminishing
What's the big picture?

Convergence

- Linear: $\text{sc} \& L_c \text{ grad, steps: const or backtracking LS}$
- Sublinear $O(1/k)$ : $w_c, L_c \text{ grad, steps: const or backtracking}$

- Sublinear $O(1/k)$: $sc$, steps: $O(1/k)$
- Sublinear $O(1/\sqrt{k})$ : $w_c$, steps $O(1/\sqrt{k})$, avg. $x$

- Sublinear $O(1/k)$: $sc$, steps: $O(1/k)$, $x_{best}$
- Sublinear $O(1/\sqrt{k})$ : $w_c$, steps $O(1/\sqrt{k})$, $x_{best}$

- Sublinear $O(1/k)$: $sc$, steps: $O(1/k)$
- Sublinear $O(1/\sqrt{k})$ : $w_c$, steps $O(1/\sqrt{k})$, avg. $x$
No. of Iterations?

To obtain eps-suboptimal solutions

• Linear convergence: \( \mathcal{O}(\log(1/\epsilon)) \)

• Sublinear with \( \mathcal{O}(1/k) : \mathcal{O}(1/\epsilon) \)

• Sublinear with \( \mathcal{O}(1/\sqrt{k}) : \mathcal{O}(1/\epsilon^2) \)
Ex. Perceptron [Rosenblatt, 1957]

Training data:  \( \{(x_j, y_j)\}_{j=1}^{m}, \ x_j \in \mathbb{R}^n, \ y_j \in \mathbb{R} \)

Prediction:  \( h(x) = \sum_{i=1}^{n} w_i x_i \)

Algorithm:

- Choose a “learning rate” 0 < r < 1
- For k=1,2,…
  - Choose an example index j
  -  \( w_{t+1} = w_t + r(y_j - h(x_j))x_j \)
Ex. Perceptron

\[ w_{t+1} = w_t + r(y_j - h(x_j))x_j \]

\[ f(w) = \mathbb{E}_{(x,y)}[(y - w^T x)^2 / 2] \]

\[ F(w; x, y) = \frac{1}{2}(y - w^T x)^2 \]

\[ \nabla F(w; x, y) = -(y - w^T x)x \]

This is stochastic gradient descent for linear regression!