Optimization - Algorithm Summary

Thu Nguyen

June 26, 2020

Contents

1 Unconstrained Optimization

Consider the problem

$$\min_{x \in \mathbb{R}^d} \; F(x)$$

where F is some *objective function*, not necessarily *convex*. We note that the *convexity* condition often helps guaranteeing the convergence of the algorithms.

We will assume that F is sufficiently smooth, ie. F is differentiable, at least once. Given f, we denote by ∇F the gradient of F.

Let $x^* = \arg\min_{x \in \mathbb{R}^d} F(x)$, ie. x^* is the (global) optimal minimer. We remark that while it is desirable to precisely get to x^*, it is often infeasible (or inefficient) to do so with numerical optimization. We instead have a number of alternating choices for the convergence condition:

1. The change in F, $F\left(x^{(t+1)}\right) - F\left(x^t\right)$ is small enough, usually less than some predefined ε.
2. The norm of the gradient $\left\|\nabla F\left(x^{(t)}\right)\right\|$ is small enough (we recall that $\nabla F(x^*) = 0$, and hence the norm is 0 at the optimal x^*.
3. Predefining the number of iterations.
4. Etc.

Hence, in the specification of the algorithms, *convergence condition* usually refers to one or more of those conditions being satisfied.

1.1 Gradient Descent

Algorithm 1 Gradient Descent

1: Initialize: $x^{(0)} \in \mathbb{R}^d$
2: **for** $t = 0, 1, \ldots$ **do**:
3: Update

$$x^{(t+1)} \;=\; x^{(t)} - \alpha_t \nabla F(x^{(t)})$$

4: until convergence

where α_t is the *step size*, which can be fixed (which in turn does not guaranteed convergence even if f is convex, but does not require any computing) or updated with *backtracking line search*.

1.1.1 Stochastic Gradient Descent

Stochastic gradient descent is a much more efficient (cheap to implement) variant of the base *gradient descent*. Suppose there are N data points, let f_i be the loss function associated with the i^{th} data point. This gives us

$$F(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$$

ie. F can be thought of as an aggregate of individual losses. Let us denote by $f_i(x^{(t)}$ be the loss from the i^{th} data point when evaluated at $x^{(t)}$.

We note that at each iteration t, f_i is sampled uniformly randomly from all the $\{f_i\}_{i=1}^{N}$.

Algorithm 2 Stochastic Gradient Descent

1: Initialize: $x^{(0)} \in \mathbb{R}^d$
2: **for** $t = 0, 1, \ldots$ **do**:
3: Sample f_i randomly from $\{f_i\}_{i=1}^{N}$, update

$$x^{(t+1)} \;=\; x^{(t)} - \alpha_t \nabla f_i(x^{(t)})$$

4: until convergence

We note that at each iteration, SGD updates based on only 1 data point, compared to all the N data points in the GD. An implication is that it is not guaranteed that the loss function F is always decreasing, although it should nevertheless is on a downward sloping trend.

1.1.2 Minibatch Gradient Descent

We observe that although *stochastic gradient descent* generally works as expected, the process can be sensitive to the random sampling of the data points. *Minibatch gradient descent* handles that problem by sampling some k data points at each iteration, versus just 1 in SGD. In practice, k is often chosen to be 16 or 32.

This gives stability to the process while not introducing significantly more computation cost.

Algorithm 3 Minibatch Gradient Descent

1: Initialize: $x^{(0)} \in \mathbb{R}^d$
2: **for** $t = 0, 1, \ldots$ **do**:
3: Sample k f_{i_j} randomly from $\{f_i\}_{i=1}^{N}$, update

$$x^{(t+1)} \;=\; x^{(t)} - \alpha_t \sum_{j=1}^{k} \nabla f_{i_j}(x^{(t)})$$

4: until convergence

1.2 Variants of Gradient Descent

1.2.1 Gradient Descent with Momentum

1.2.2 Gradient Descent with Nesterov Acceleration

1.3 Conjugate Gradient

2 Constrained Optimization

Consider the original problem of minimizing some objective function f, but this time subject to additional conditions

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad \text{for } i = 1, \ldots, m \\ & h_i(x) = 0, \quad \text{for } i = 1, \ldots, p \end{aligned} \tag{P}$$

Let us consider the *Lagrangian function* $L : \mathbb{R}^d \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ such that

$$L(x, \lambda, \nu) \;=\; f(x) + \left(\sum_{i=1}^{m} \lambda_i g_i(x) \right) + \left(\sum_{i=1}^{p} \nu_i h_i(x) \right)$$

where $\lambda \in \mathbb{R}^m$ and $\nu \in \mathbb{R}^p$ are the Lagrangian multipliers. We define the *Lagrange dual function* $\hat{f} : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ as

$$\hat{f}(\lambda, \nu) \;=\; \inf_{x \in \mathbb{R}^d} L(x, \lambda, \nu).$$

We remark that $\hat{f}$ is *concave*, regardless of whether or not the original function f is convex.

Solving the dual function gives us the lower bounds on the the optimal x^*, ie. let $\tilde{x}$ be a feasible point (a point which satisfies all the g_i and h_i conditions), then

$$\forall \tilde{x} : \quad \hat{f}(\lambda, \nu) \;=\; \inf_{x \in \mathbb{R}^d} L(x, \lambda, \nu) \;\leq\; L(\tilde{x}, \lambda, \nu) \;\leq\; f(\tilde{x})$$

Under appropriate conditions (for example the KKT conditions), we observe the desired equality

$$\hat{f}(\lambda, \nu) \;=\; f(\tilde{x})$$

which implies that by solving solving the unconstrained dual problem

$$\max_{\lambda, \nu} \quad \hat{f}(\lambda, \nu),$$

we will have simultaneously solved the prior problem in (P).

2.1 Dual Ascent

2.2 Method of Multipliers

2.3 Alternating Direction Method of Multipliers